

UNIVERSIDADE FEDERAL DO PARANÁ

CIRO ALEXANDRE STAHLSCMIDT CORRÊA  
JUAN NICOLAS VARELA

**REDE SOCIAL *VOLUNTARIUS***

CURITIBA

2018

CIRO ALEXANDRE STAHLSCMIDT CORRÊA  
JUAN NICOLAS VARELA

**REDE SOCIAL *VOLUNTARIUS***

Trabalho apresentado à disciplina de Conclusão do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional Tecnológica, da Universidade Federal do Paraná.

Orientador: Prof. Razer Anthom Nizer Rojas Montaña

CURITIBA  
2018

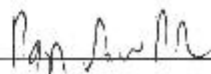
## TERMO DE APROVAÇÃO

Ciro Alexandre Stahlschmidt Correa

Juan Nicolas Varela

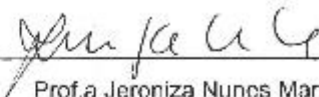
## REDE SOCIAL VOLUNTARIUS

Monografia aprovada como requisito parcial à obtenção do título de  
Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação  
Profissional e Tecnológica da Universidade Federal do Paraná.



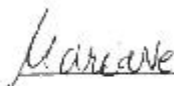
Prof. Razer Anthom Nizer Rojas Montaño

Orientador – SEPT/UFPR



Prof.ª Jeroniza Nunes Marchaukoski

SEPT/UFPR



Mariane Gonçalves Kulik

SEPT/UFPR

Curitiba, 04 de Dezembro de 2018.

Dedicamos esse trabalho à  
Universidade Federal do Paraná, em  
especial ao Setor de Educação  
Profissional e Tecnologia e seu corpo  
Docente.

## **AGRADECIMENTOS**

Gostaríamos de deixar aqui registrado nossos sinceros agradecimentos a pessoas que de uma forma ou de outra contribuíram para que realizássemos não só esse projeto, mas toda a caminhada ao longo do curso.

Agradecemos aos professores pela transmissão do conhecimento, contribuindo para nossa formação e crescimento intelectual. Em especial, aos professores Alexander Robert Kutzke, Andreia de Jesus, Jaime Wojciechowski, Jeroniza Nunes Marchaukoski, Luiz Antonio Passos Cardoso, Luiz Antonio Pereira Neves, Mauro Antonio Alves Castro, Rafael Romulado Wandresen, Rafaela Mantovani Fontana e Razer Anthom Nizer Rojas Montaña, pela amizade, didática e atenção dispendida nas dúvidas que surgiam durante as aulas. Aos professores Maria Valéria da Costa e Paulo Eduardo Sobreira Moraes, por trazer conhecimento com qualidade da área de psicologia e empreendedorismo.

A todos os colaboradores do Setor de Educação Profissional Profissional e Tecnológica, por manter e cuidar das dependências, nos orientar e nos receber bem, toda vez que precisamos resolver algum assunto ou esclarecer alguma dúvida.

Aos nossos familiares, em especial mães, esposa, filha e irmãos, e também nossos amigos, por compreenderem nossas ausências durante as horas de estudo. Aos nossos gestores, pelas concessões realizadas ao longo do curso.

À Deus, por nos conceder, saúde, inteligência, força de vontade, paciência, humildade e colocar as pessoas certas em nosso caminho, permitindo que concluíssemos esse curso de graduação. A todos, nosso muito obrigado!

“O generoso sempre prosperará; quem  
oferece ajuda ao necessitado, conforto receberá.”  
(Provérbios 11:25)

## RESUMO

O presente trabalho aborda o desenvolvimento de um software para ser utilizado através da internet, no formato de uma rede social, com o propósito de facilitar a aproximação entre aqueles que precisam de ajuda e pessoas dispostas a ajudar, através da dedicação do seu tempo, principalmente. As oportunidades de voluntariado são realizadas na forma de *posts*, que podem ser curtidos e comentados pelos usuários, possibilitando a comunicação em duas vias e gerando maior interação entre os envolvidos. Os usuários relacionam-se a partir de um vínculo de amizade e em seu perfil relacionam causas que abraçam e, no caso de voluntários, habilidades que possuem para contribuir, facilitando assim a oferta e a procura. Para o desenvolvimento do software, foi utilizada a metodologia *Scrum*, que divide o trabalho em entregas semanais contendo funcionalidades visíveis e utilizáveis por parte do usuário. O software foi especificado utilizando os principais conceitos da UML e desenvolvido utilizando linguagem orientada a objetos Java e banco de dados relacional MySQL, aliado a outras tecnológias.

**Palavras-chave:** Voluntariado. Voluntário. Rede Social. *Scrum*. Java. UML. MySQL.

## ABSTRACT

The present work addresses the development of a software to be used through the internet, in the form of a social network, with the purpose of facilitating the rapprochement between those who need help and people willing to help, through the dedication of their time, mainly. Volunteer opportunities are held in the form of posts, which can be tanned and commented by users, enabling two-way communication and generating greater interaction among those involved. Users relate to each other through a bond of friendship and in their profile relate causes they embrace and, in the case of volunteers, the skills they have to contribute, thus facilitating supply and demand. For the development of the software, the Scrum methodology was used, which divides the work into weekly deliveries containing visible and usable user functionalities. The software was specified using the main concepts of UML and developed using Java Oriented-object programming and MySQL relational database, along with other technologies.

**Key-words:** Volunteering. Volunteer. Social Network. Scrum. Java. UML. MySQL.



## LISTA DE FIGURAS

FIGURA 1 - PESQUISA IBGE.....	13
FIGURA 2 - EXEMPLO DE ESTRUTURA.....	24
FIGURA 3 - CLIENTE X SERVIDOR WEB .....	27
FIGURA 4 - LINK ENTRE HTML E JAVASCRIPT .....	28
FIGURA 5- CSS INCORPORADO AO HTML .....	29
FIGURA 6 - LINK ENTRE HTML E CSS .....	30
FIGURA 7 - EXEMPLO DE CLASSES.....	32
FIGURA 8 - CÓDIGO-FONTE, BYTECODES E CÓDIGO NATIVO.....	33
FIGURA 9 - PADRÃO MVC.....	35
FIGURA 10 - REPRESENTAÇÃO GRÁFICA: CLASSE USUÁRIO .....	37
FIGURA 11 - REPRESENTAÇÃO GRÁFICA: CASO DE USO REALIZAR CADASTRO .....	38
FIGURA 12 - ITENS COMPORTAMENTAIS, DE AGRUPAMENTO E ANOTACIONAIS .....	39
FIGURA 13 - ASSOCIAÇÃO DE CLASSES.....	40
FIGURA 14 - GENERALIZAÇÃO .....	41
FIGURA 15 - METODOLOGIA CLÁSSICA X METODOLOGIA ÁGIL .....	43
FIGURA 16 - ENGENHARIA DE SISTEMA .....	45
FIGURA 17 - ANÁLISE DE REQUISITOS.....	46
FIGURA 18 - PROJETO TÉCNICO.....	47
FIGURA 19 - CONSTRUÇÃO .....	48
FIGURA 20 - VALIDAÇÃO .....	49
FIGURA 21 - CICLO DO SCRUM .....	51
FIGURA 22 - EXEMPLO DE BACKLOG .....	52
FIGURA 23 - PACOTES .....	60
FIGURA 24 - INTERFACE GRÁFICA WBSTOOL.....	65
FIGURA 25 - INTERFACE GRÁFICA TEAMGANTT .....	66
FIGURA 26 - INTERFACE GRÁFICA ASTAH.....	67
FIGURA 27 - INTERFACE GRÁFICA brMODELO.....	68
FIGURA 28 - INTERFACE GRÁFICA NETBEANS - BIBLIOTECAS.....	70
FIGURA 29 - INTERFACE GRÁFICA MYSQL WORKBENCH .....	71
FIGURA 30 - ESCOLHA DO GLASSFISH .....	72

FIGURA 31 - ARQUITETURA.....	73
FIGURA 32 - PÁGINA INICIAL.....	74
FIGURA 33 - PÁGINA DE ACESSO / CADASTRO .....	75
FIGURA 34 - TIMELINE .....	76
FIGURA 35 - PERFIL VOLUNTÁRIO .....	77
FIGURA 36 - MINHAS INSTITUIÇÕES.....	77
FIGURA 37 - MEUS AMIGOS .....	78
FIGURA 38 - PERFIL INSTITUIÇÕES .....	79
FIGURA 39 - MEUS VOLUNTÁRIOS.....	79
FIGURA 40 - CRIAR OPORTUNIDADE.....	80
FIGURA 41 - PESQUISAR VOLUNTÁRIOS .....	81
FIGURA 42 - PESQUISAR INSTITUIÇÕES.....	81
FIGURA 43 - PESQUISAR OPORTUNIDADES.....	82
FIGURA 44 - VISUALIZAR OPORTUNIDADE .....	83
FIGURA 45 - LISTAR VOLUNTÁRIOS .....	84
FIGURA 46 - EDITAR VOLUNTÁRIO .....	84
FIGURA 47 - CERTIFICADO .....	85
FIGURA 48 - ESTRUTURA ANALÍTICA DE PROJETO .....	95
FIGURA 49 - CRONOGRAMA DO PROJETO .....	96
FIGURA 50 - DIAGRAMA DE CASO DE USO .....	97
FIGURA 51 - DIAGRAMA DE CLASSES .....	142
FIGURA 52 – DIAGRAMA DE CLASSE PACOTES.....	143
FIGURA 53 – DIAGRAMA DE CLASSE VIEWS .....	143
FIGURA 54 – DIAGRAMA DE CLASSE CONTROLLER .....	144
FIGURA 55 – DIAGRAMA DE CLASSE FAÇADE .....	144
FIGURA 56 – DIAGRAMA DE CLASSE DAO .....	145
FIGURA 57 - EFETUAR CADASTRO .....	146
FIGURA 58 - EFETUAR ACESSO .....	146
FIGURA 59 - MANTER PERFIL INSTITUIÇÃO .....	147
FIGURA 60 - MANTER PERFIL VOLUNTÁRIO.....	147
FIGURA 61 - MEUS AMIGOS .....	148
FIGURA 62 - MEUS VOLUNTÁRIOS.....	148
FIGURA 63 - MINHAS INSTITUIÇÕES.....	149
FIGURA 64 - OPORTUNIDADE – VISUALIZAR .....	149

FIGURA 65 - OPORTUNIDADE – EDITAR.....	150
FIGURA 66 - OPORTUNIDADE - LISTAR VOLUNTÁRIOS.....	150
FIGURA 67 - OPORTUNIDADE – EDITAR VOLUNTÁRIO.....	151
FIGURA 68 - PESQUISAR INSTITUIÇÕES.....	151
FIGURA 69 - PESQUISAR OPORTUNIDADES.....	152
FIGURA 70 - PESQUISAR PESSOAS.....	152
FIGURA 71 - TIMELINE .....	153
FIGURA 72 – CHAT .....	154
FIGURA 73 – MANTER OPORTUNIDADE .....	155
FIGURA 74 – MANTER AMIZADE.....	156
FIGURA 75 - DIAGRAMA DE ENTIDADE RELACIONAMENTO .....	157

## LISTA DE QUADROS

QUADRO 1 - PESQUISA DE REDES SOCIAIS DE VOLUNTARIADO .....	23
QUADRO 2 - TABELA DE USUÁRIOS CADASTRADOS.....	25
QUADRO 3 - COMANDOS SQL .....	31
QUADRO 4 - RECURSOS DO PROJETO .....	56
QUADRO 5 - BACKLOG DO PRODUTO .....	57
QUADRO 6 - RISCOS DO PROJETO .....	58
QUADRO 7 - MATERIAL ADQUIRIDO .....	66
QUADRO 8- HARDWARE.....	72

## LISTA DE ABREVIATURAS E SIGLAS

ABNT	- Associação Brasileira de Normas Técnicas
AJAX	- <i>Asynchronous JavaScript and XML</i>
ANSI	- <i>American National Standards Institute</i>
CNPJ	- Cadastro Nacional de Pessoa Jurídica
CPF	- Cadastro de Pessoa Física
CSS	- <i>Cascading Style Sheets</i>
DAO	- <i>Data Access Object</i>
DDL	- <i>Data Definition Language</i>
DML	- <i>Data Manipulation Language</i>
EAP	- Estrutura Analítica de Projeto
ECMA	- European Computer Manufactures Association
EE	- <i>Enterprise Edition</i>
EJB	- <i>Enterprise Java Beans</i>
EL	- <i>Expression Language</i>
ERP	- <i>Enterprise Resource Planning</i>
GIFE	- Grupo de Institutos Fundações e Empresas
GPL	- <i>General Public Licence</i>
HTML	- <i>Hyper Text Markup Language</i>
IBGE	- Instituto Brasileiro de Geografia e Estatística
IBM	- <i>International Business Machines</i>
IDE	- <i>Integrated Development Environment</i>
IED	- <i>Ideal Engineering Day</i>
IMS	- <i>Information Management System</i>
INC	- <i>Incorporation</i>
ISO	- <i>International Standard Organization</i>
JSP	- <i>Java Server Pages</i>
JSTL	- <i>JSP Standard Tag Library</i>
JVM	- <i>Java Virtual Machine</i>
LTDA	- Limitada
ME	- <i>Micro Edition</i>
MEC	- Ministério da Educação e Cultura
MT	- Mato Grosso

MVC	- <i>Model-View-Controller</i>
noSQL	- <i>Not Only Structured Query Language</i>
OMT	- <i>Object Modeling Technique</i>
ONG	- Organizações Não-Governamentais
ONU	- Organização das Nações Unidas
OOSE	- <i>Object Oriented Software Engineering</i>
OSCIP	- Organização da Sociedade Civil de Interesse Público
PHP	- <i>Hypertext Preprocessor</i>
ROI	- <i>Return On Investment</i>
SC	- Santa Catarina
SE	- <i>Standard Edition</i>
SGBD	- Sistema Gerenciador de Banco de Dados
SQL	- <i>Structured Query Language</i>
SSO	- <i>Single Sign-On</i>
TCC	- Trabalho de Conclusão de Curso
UML	- <i>Unified Modeling Language</i>
XP	- <i>Extreme Programming</i>
WBS	- <i>Work Breakdown Structure</i>

## LISTA DE SÍMBOLOS

™ - trade mark

® - marca registrada

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>13</b>
1.1 PROBLEMA .....	14
1.2 OBJETIVO GERAL .....	15
1.3 OBJETIVOS ESPECÍFICOS .....	15
1.4 JUSTIFICATIVA .....	15
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>16</b>
2.1 ASPECTOS CONCEITUAIS DO NEGÓCIO .....	16
2.1.1 Terceiro Setor.....	16
2.1.2 Organizações .....	19
2.1.3 Trabalho Voluntário .....	20
2.1.4 Causas .....	21
2.1.5 Habilidades.....	22
2.1.6 Rede Social .....	22
2.2 ASPECTOS CONCEITUAIS DA TECNOLOGIA ENVOLVIDA.....	24
2.2.1 Sistema Gerenciador de Banco de Dados .....	24
2.2.2 Linguagens de Programação .....	26
2.2.3 Modelagem - UML.....	36
2.2.4 Gerenciamento de Projetos.....	42
<b>3 MATERIAIS E MÉTODOS .....</b>	<b>55</b>
3.1 APLICAÇÃO DO SCRUM E ADAPTAÇÕES.....	55
3.1.1 Pré-Game.....	55
3.1.2 Game.....	59
3.1.3 Pós-Game .....	64
3.2 MATERIAIS UTILIZADOS .....	64
3.2.1 WBSTool .....	65
3.2.2 Teamgantt .....	65
3.2.3 Template Redes Sociais.....	66
3.2.4 Astah .....	66
3.2.5 brModelo .....	67
3.2.6 Suítes Google e Microsoft .....	68
3.2.7 Netbeans .....	69
3.2.8 MySQL .....	70

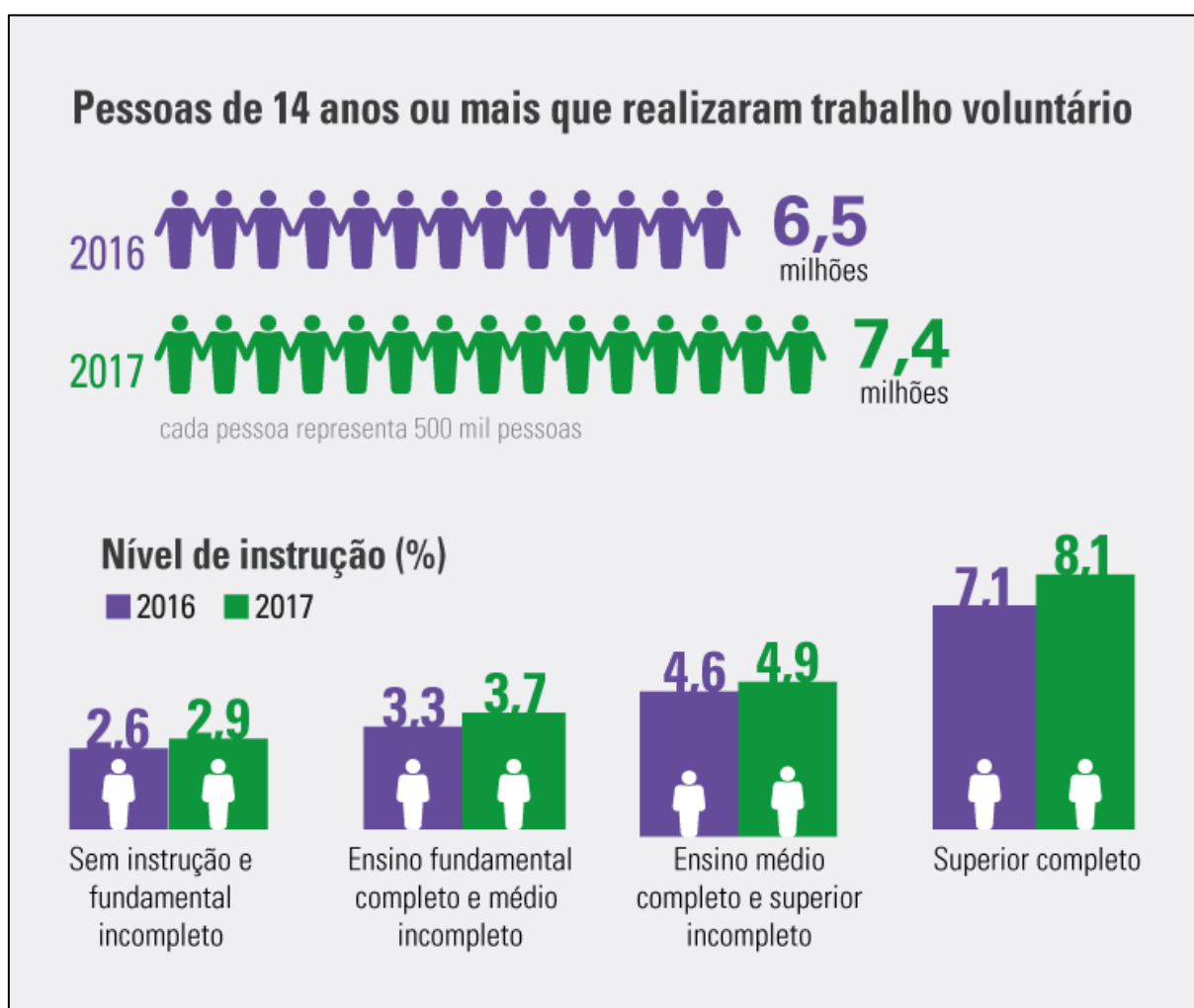


3.2.9 Glassfish.....	71
3.2.10 Hardware.....	72
<b>4 APRESENTAÇÃO DO SISTEMA .....</b>	<b>73</b>
4.1 ARQUITETURA DO SOFTWARE .....	73
4.2 SOFTWARE .....	74
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>86</b>
5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS .....	87
5.1.1 Aplicativo Móvel .....	87
5.1.2 Timeline.....	88
5.1.3 Compartilhar Publicações.....	88
5.1.4 Cadastros .....	88
5.1.5 Doações .....	89
5.1.6 Segurança .....	89
<b>REFERÊNCIAS.....</b>	<b>91</b>
<b>APÊNDICE A – ESTRUTURA ANALÍTICA DE PROJETO .....</b>	<b>95</b>
<b>APÊNDICE B – CRONOGRAMA DO PROJETO .....</b>	<b>96</b>
<b>APÊNDICE C – DIAGRAMA DE CASO DE USO.....</b>	<b>97</b>
<b>APÊNDICE D – ESPECIFICAÇÕES DE CASOS DE USO .....</b>	<b>98</b>
<b>APÊNDICE E – DIAGRAMA DE CLASSES .....</b>	<b>142</b>
<b>APÊNDICE F – DIAGRAMAS DE SEQUÊNCIA .....</b>	<b>146</b>
<b>APÊNDICE G - DIAGRAMA DE ESTADO .....</b>	<b>155</b>
<b>APÊNDICE H - DIAGRAMA DE ENTIDADE RELACIONAMENTO .....</b>	<b>157</b>

## 1 INTRODUÇÃO

O trabalho voluntário teve a participação de 7,4 milhões de pessoas no ano de 2017, 840 mil pessoas a mais do que no ano de 2016, segundo pesquisa publicada pela agência de notícias do IBGE, em 18/04/2018, que revela também um aumento da prática do voluntariado conforme o nível educacional. Na categoria de pessoas sem instrução ou com fundamental incompleto, apenas 2,9% faziam trabalho voluntário, enquanto na categoria de pessoas com superior completo, 8,1% realizaram essa atividade, conforme ilustrado pela FIGURA 1 (IBGE, 2018).

FIGURA 1 - PESQUISA IBGE



FONTE: IBGE (2018).

Outro fato relevante, ocorrido no ano de 2018, foi a homologação de uma diretriz que autoriza a inclusão de horas de trabalho voluntário em históricos acadêmicos de escolas e de universidades, pelo presidente da República, Michel

Temer e pelo ministro da Educação, Rossieli Soares, em cerimônia em comemoração ao Dia Nacional do Voluntariado, segundo publicação no portal do ministério da Educação e Cultura. De acordo com a publicação, as atividades curriculares ligadas ao voluntário poderão ser acrescentadas à carga-horária mínima, por iniciativa da instituição de ensino, e ser contadas a mais, inclusive, no histórico escolar do aluno. Para a educação básica, as regras são um pouco diferentes, pois essas horas não serão computadas à carga-horária mínima, sendo consideradas extraordinárias, ou seja, acessórias, aditivas e complementares ao conteúdo curricular mínimo obrigatório exigido pela regulação específica (MEC, 2018).

Segundo Soares (2018), a importância dessa resolução homologada é de trazer o voluntariado para dentro das instituições de ensino, que complementado por Padilha (2018), contribui para o desenvolvimento integral dos educandos e a articulação com as comunidades locais e o entorno escolar. Dessa forma, o número de pessoas participando em ações de voluntariado tenderá a aumentar nos próximos anos.

## 1.1 PROBLEMA

Para que as pessoas se interessem e participam cada vez mais de ações de voluntariado, é importante que estas tenham maior visibilidade e alcance junto à sociedade como um todo. Neste contexto, a tecnologia pode ser uma aliada para viabilizar esse fluxo de informações, aproximando voluntários e oportunidades de voluntariado.

Pelo fato de o público jovem ter facilidade e estar constantemente conectado à internet, a criação de uma rede social voltada para o tema, pode somar e contribuir para a realização dessa aproximação. Além disso, a possibilidade de obter o certificado de forma *online*, ao concluir a realização de uma atividade de voluntariado, foi um fator motivacional à parte, uma vez que a obtenção desse documento é burocrática e ocorre geralmente após o voluntário ter concluído a atividade na instituição.

## 1.2 OBJETIVO GERAL

Como objetivo geral, o intuito desse trabalho é desenvolver uma rede social destinada a ações de voluntariado, onde voluntários e instituições possam se comunicar em uma via de duas mãos, permitindo uma maior participação e envolvimento de todos nela cadastrados.

## 1.3 OBJETIVOS ESPECÍFICOS

Em se tratando de objetivos específicos, pode-se elencar os seguintes itens:

- 1) Voluntários poderem ler, curtir, comentar, e criar *posts*;
- 2) Instituições publicarem oportunidades de voluntariado e registrar a quantidade de horas trabalhadas pelo voluntário;
- 3) Instituições emitirem o certificado de participação *online*, facilitando dessa maneira a entrega do documento ao voluntário;

## 1.4 JUSTIFICATIVA

No entendimento da equipe, a criação de uma rede social com os objetivos acima listados vai de encontro com os objetivos traçados pelo ministério da educação, que procura atrair estudantes a participarem de forma mais ativa, através de suas escolas e universidades. A utilização de redes sociais para atrair não só esse tipo de público, como o restante da sociedade, poderia contribuir nesse objetivo, e coincidentemente, após o início desse trabalho, o governo lançou, no segundo semestre desse ano, uma plataforma de voluntariado com esse objetivo. Outro ganho que a plataforma traz, é a possibilidade das instituições acompanharem as necessidades umas das outras, compartilhando experiências e até mesmo soluções, caso as tenham encontrado.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo, são apresentados os aspectos conceituais envolvidos na elaboração desse trabalho.

Na primeira seção, são explicados os conceitos que envolvem o negócio em si, abordando o conceito de terceiro setor, agentes envolvidos, causas, trabalho voluntário e habilidades.

Na segunda seção, são explicados os conceitos que envolvem toda a tecnologia utilizada para a criação do software desenvolvido, seus conceitos e suas características.

### 2.1 ASPECTOS CONCEITUAIS DO NEGÓCIO

A partir do momento que se decidiu desenvolver uma rede social de ações de voluntariado, com o objetivo de facilitá-las através da aproximação entre voluntários e entidades, foi necessário se aprofundar no assunto e descobrir uma vasta quantidade de informações. Aspectos jurídicos, contábeis, políticos, econômicos, bem como formas de administração em si, são exemplos de informações encontradas durante a pesquisa.

O objetivo aqui é limitar-se a trazer alguns conceitos considerados relevantes, de forma a facilitar o entendimento do negócio e com isso deixar mais clara a compreensão do software desenvolvido.

#### 2.1.1 Terceiro Setor

O termo passou a ser utilizado no Brasil com mais ênfase a partir do ano de 1996, quando ocorreu o III Encontro Ibero-Americano do Terceiro Setor, segundo. Os encontros anteriores, realizados na Espanha e México, eram denominados Encuentro Iberoamericano de Filantropia, e a mudança do nome foi um dos principais assuntos discutidos durante sua realização no Rio de Janeiro, com participação do GIFE – Grupo de Institutos, Fundações e Empresas (IOSCHPE, 2005).

Traduzida do inglês, *third sector*, a expressão é integrante do vocabulário sociológico nos Estados Unidos, segundo Fernandes (2005) e diretamente relacionada a outras duas, organizações sem fins lucrativos (*non profit organizations*)

e organizações voluntárias (*volunteer organizations*). Para a maioria dos americanos, o Primeiro Setor seria o mercado e o Segundo Setor seria o Governo, embora a classificação contrária também seja encontrada. Logo, por exclusão, o Terceiro Setor seria então o ramo da sociedade cujas ações visam o interesse público, porém sem fins lucrativos (COELHO, 2002).

Aqui no Brasil, a adoção do termo tem como pano de fundo todo um histórico de mobilizações realizadas pela sociedade, cujo início remonta à época da ditadura. Com o golpe militar em 1964 e a publicação de uma nova Constituição em 1967 legalizando a ditadura, a sociedade civil se organiza em protestos e passeatas contra o regime, sendo coibida com a publicação do Ato Institucional número 5, que endurece ainda mais o regime e suspende os direitos democráticos (VILLELA, 2015).

Durante esse período, o termo ONG (organizações não-governamentais) surge, advindo da Europa, através de estímulo da ONU para que países do Primeiro Mundo ajudem no desenvolvimento de países do Terceiro Mundo. Mesmo designando sua natureza não-governamental, o termo no Brasil está mais associado a um tipo de organização, distinguindo-se de institutos e fundações, por exemplo. Entretanto, a expressão “sociedade civil” se torna mais abrangente devido ao fato das lutas contra o regime militar, bem como, após o término do regime, se manifestar evidenciando sua desvinculação do Estado (FERNANDES, 2005).

Com a publicação da anistia em 1979 e o fim da ditadura em 1985, a sociedade civil começa a ocupar mais espaço e isso é percebido através de manifestações como o “Diretas Já” e o “Impeachment”, por exemplo. Além disso, a participação na concretização do Estatuto da Criança e do Adolescente e o surgimento de iniciativas como os programas Ação da Cidadania, criado por Herbert de Souza, o Betinho, e a Pastoral da Criança, criada por Zilda Arns, são exemplos, entre outros, da mobilização da sociedade civil, que passa a perceber sua cota de responsabilidade para que o País funcione, responsabilidade essa que era dividida entre o Estado e o mercado (VILLELA, 2015).

Em 1995, Ruth Cardoso, a primeira dama, lança o programa Comunidade Solidária, através de parcerias entre organizações não-governamentais, universidades, empresas e governos, com o objetivo de construir e difundir programas sociais inovadores, e ao participar do encontro citado no início, declara:

Gostaria, inicialmente, de realçar a importância que atribuo a este Encontro Ibero-Americano do Terceiro Setor e declarar meu apoio a esta iniciativa. Foi a crença na existência, relevância e potencialidade de um Terceiro Setor, não-lucrativo e não-governamental, que inspirou o desenho do programa que coordeno através do Conselho da Comunidade Solidária. Tudo isso é novo e está se configurando debaixo de nossos olhos. O próprio conceito de Terceiro Setor, seu perfil e contornos ainda não estão claros nem sequer para muitos dos atores que o estão compondo. São vários os termos que temos utilizado para caracterizar este espaço que não é Estado nem mercado e cujas ações visam ao interesse público: iniciativas sem fins lucrativos, filantrópicas, voluntárias (CARDOSO, 2005).

Fica evidente no discurso que conceituar Terceiro Setor se torna uma tarefa polêmica, com diferenças entre alguns autores, como será visto a seguir. Talvez esteja relacionado ao fato de não haver uma definição explícita do que ele é, mas só haver uma definição do que ele não é: nem estado e nem mercado.

Para Fernandes, o Terceiro Setor é:

Composto de organizações sem fins lucrativos, criadas e mantidas pela ênfase na participação voluntária, num âmbito não- governamental, dando continuidade às práticas tradicionais da caridade, da filantropia e do mecenato e expandindo o seu sentido para outros domínios, graças, sobretudo, à incorporação do conceito de cidadania e de suas múltiplas manifestações na sociedade civil (FERNANDES, 2005).

Outras definições são listadas na obra de Dias (2008, 103 – 104), que realizou um estudo do ponto de vista jurídico sobre o Terceiro Setor, consolidando diversas opiniões e abordagens sobre o tema. De maneira simples e objetiva, Thompson (2005 citado por DIAS, 2008) define Terceiro Setor como constituído por “todas aquelas instituições sem fins lucrativos que, a partir do âmbito privado, perseguem propósitos de interesse público”.

Merege (2001 citado por DIAS, 2008) não só define, determina a forma, como se origina, seus objetivos, sua finalidade (não lucrativa) e com quais recursos realiza sua atividade:

O terceiro setor constitui-se de organizações criadas por iniciativas de cidadãos com o objetivo de prestar serviços ao público sem fins lucrativos (saúde, educação, cultura, habitação, direitos civis, desenvolvimento do ser humano, proteção ao meio ambiente), ainda que eventuais excedentes sejam reaplicados na manutenção das próprias atividades ou distribuídos entre os colaboradores. Suas receitas podem ser geradas em atividades operacionais, mas resultam sobretudo de doações do setor privado ou do setor governamental (DIAS, 2008).

Na legislação brasileira, não se encontra definição a respeito de Terceiro Setor, mas sim um conjunto de legislações que utilizam os termos Organizações

Sociais, Organização da Sociedade Civil de Interesse Público (OSCIP), utilidade pública, entidades de fins filantrópicos, entidades beneficentes de assistência social e cooperativa social.

### 2.1.2 Organizações

Segundo Chiavenato (2014, p. 24), organização é uma entidade social composta de pessoas e recursos deliberadamente estruturada e orientada para alcançar um objetivo comum.

As organizações do Terceiro Setor são caracterizadas principalmente pelo papel econômico que possuem, que é não objetivar o lucro, além de atender alguma necessidade de serviço público. Porém, somente essas duas classificações não são suficientes, é necessário que sejam organizações privadas também, o que as diferencia das empresas públicas (COELHO, 2002, p. 59-60).

Dias (2008) apresenta a seguinte qualificação jurídica atribuída às entidades privadas sem fins lucrativos integrantes do Terceiro Setor:

- a. Declaração de Utilidade Pública: “As Sociedades Civis, as Associações e as fundações constituídas no país com o fim exclusivo de servir desinteressadamente à coletividade [...]” (art. 1º, da Lei nº 91, de 28 de agosto de 1935;
- b. Certificado de entidade de fins filantrópicos: “Considera-se entidade beneficente de assistência social a pessoa jurídica de direito privado, sem fins lucrativos, que atue no sentido de: I – proteger a família, a maternidade, a infância, a adolescência e a velhice; II – amparar crianças e adolescentes carentes; III – promover ações de prevenção, habilitação e reabilitação de pessoas portadoras de deficiências; IV – promover, gratuitamente, assistência educacional ou de saúde; V – promover a integração ao mercado de trabalho.” (art. 2º, Decreto nº 2.536, 6 de abril de 1998);
- c. Organização Social: “[...] pessoas jurídicas de direito privado, sem fins lucrativos, cujas atividades sejam dirigidas ao ensino, à pesquisa científica, ao desenvolvimento tecnológico, à proteção e preservação do meio ambiente, à cultura e à saúde.” (art. 1º, da Lei nº 9.637, 15 de maio de 1998);
- d. Organização da Sociedade Civil de Interesse Público: “[...] pessoas jurídicas de direito privado, sem fins lucrativos, desde que os respectivos objetivos



sociais e normas estatutárias atendam aos requisitos instituídos por esta Lei.” (art. 1º e §1º Lei nº 9.790, de 23 de março de 1999).

Para este trabalho, a equipe preferiu utilizar o termo instituição e por consequência agrupá-las abaixo do termo Instituições, que segundo o dicionário *online* Michaelis, possuem, dentre outras, as seguintes definições:

1. Instituição:

- a. Ato ou efeito de instituir, estabelecer; instauração;
- b. Estrutura social, estabelecida por lei, que tem vigência em certo Estado ou para certo povo;
- c. Organização pública ou privada, regida por estatutos ou leis, cujo objetivo é satisfazer as necessidades de uma sociedade ou de uma comunidade de projeção social.

2. Instituições:

- a. Estruturas sociais estabelecidas pelas leis fundamentais de um país, especialmente as que resultam do direito público.

### 2.1.3 Trabalho Voluntário

O trabalho voluntário é regulamentado no Brasil pela Lei 9.608/1998, cujo artigo 1º traz a seguinte definição:

Considera-se serviço voluntário, para os fins desta Lei, a atividade não remunerada prestada por pessoa física a entidade pública de qualquer natureza ou a instituição privada de fins não lucrativos que tenha objetivos cívicos, culturais, educacionais, científicos, recreativos ou de assistência à pessoa (BRASIL, 1998).

Comemorado todo ano no Brasil, no dia 28 de agosto, e instituído pela ONU, no dia 5 de dezembro, em 1985, a atividade mobiliza milhões de pessoas no mundo inteiro. Só no Brasil, cerca de 6 milhões de pessoas realizaram esse tipo de atividade no ano de 2016 (IBGE, 2016).

Segundo a ONU, voluntário é:

O jovem, adulto ou idoso que, devido a seu interesse pessoal e seu espírito cívico, dedica parte de seu tempo, sem remuneração, a diversas formas de atividades de bem-estar social ou outros campos (ONU, 2018).

Coelho (2002) entende que com isso a ação de voluntariado existe desde os primórdios da civilização humana e é incentivada por quase todas as religiões, que mobilizam as pessoas para a realização da caridade. Geralmente, as pessoas se mobilizam em trabalhar voluntariamente por razões altruísticas, podendo ser incluído também algum interesse próprio, como aquisição de experiência profissional, enriquecimento de currículo, entre outros.

#### 2.1.4 Causas

As organizações ligadas ao voluntariado geralmente possuem como causa algum problema social não resolvido ou atendido de forma precária pelo Estado. Porém, não são todas e quaisquer atividades que podem ser realizadas, as atividades relacionadas ao poder de polícia, domínio econômico, fomento e regulação, são exemplos de atividades que não podem ser atribuídas a outros que não o Estado. Já atividades de serviço público poderiam ser realizadas, o problema é que não existe um rol de atividades de forma explícita (DIAS, 2008, p. 212).

Segundo Di Pietro (2006 citada por DIAS, 2008), o serviço público social:

[...] é o que atende a necessidades coletivas em que a atuação do Estado é essencial, mas que convivem com a iniciativa privada, tal como ocorre com os serviços de saúde, educação, previdência, cultura, meio ambiente; são tratados na Constituição no capítulo da ordem social. (DIAS, 2008).

Os artigos da Constituição mencionados são, por exemplo, art. 197 e art. 199 (saúde); art. 204 (assistência social); art. 205 (educação); art. 215 (cultura); art. 218 (ciência e tecnologia); dentre outros. Segundo Dias (2008, p. 219), outras normas infraconstitucionais ampliam o leque de atividades que podem ser desenvolvidas pelas entidades do terceiro setor com o fomento do Estado: pesquisa científica, atividades educacionais ou culturais, proteção à família, à maternidade, à infância, à adolescência, e à velhice; amparo às crianças e adolescentes carentes; auxílio a pessoas portadoras de deficiência; integração ao mercado de trabalho.

### 2.1.5 Habilidades

Ao realizar algum trabalho voluntário as pessoas procuram doar seu tempo naquilo que fazem de melhor, ou seja, alguma atividade relacionada à profissão em que atua. Em alguns casos, para a realização de determinadas atividades de voluntariado, é necessário que a pessoa tenha qualificação para poder prestar o serviço sem infringir nenhuma legislação, como é o caso de serviços ligados à área de saúde, por exemplo, no caso de atendimento médico e odontológico.

Entretanto, existem trabalhos voluntários que requerem apenas a doação de tempo, como por exemplo contar histórias para crianças, realizar visitas à idosos em asilos e conversar, ouvindo suas histórias e experiências de vida, entre outras atividades.

É importante ressaltar que algumas habilidades interpessoais ajudam ao realizar voluntariado, como por exemplo, trabalhar em equipe, ter iniciativa e saber se comunicar, ter empatia e ser motivado, bem como se comprometer à causa. Essas habilidades, é claro, podem ser adquiridas com prática ao longo do tempo, o importante é realizar a ação com muito amor e dedicação ao próximo.

### 2.1.6 Rede Social

A história da humanidade revela que os seres-humanos vivem em grupos desde os tempos mais remotos. Na pré-história, apesar da prática do nomadismo, as pessoas se juntavam em bandos, por questões de sobrevivência. Na idade antiga, surgiram as cidades, agrupando uma quantidade ainda maior de pessoas, que se relacionavam de acordo com sua situação social, raça, religião ou outras formas de divisão, compartilhando valores e objetivos em comum. Segundo Giddens (2012, p. 578), os sociólogos denominam essas conexões de redes – conexões diretas e indiretas que ligam uma pessoa ou grupo a outras pessoas ou grupos.

Não diferente, após a criação da Internet, também se percebe a criação de ferramentas expandindo ainda mais a forma como as pessoas interagem umas com as outras, aproveitando os recursos disponibilizados pela tecnologia. Exemplo disso são os aplicativos de comunicação instantânea, o e-mail, as páginas de internet, as redes sociais, entre outros (GIDDENS, 2012).

Segundo Telles (2011, p. 7), o termo “Rede Social” foi usado há cerca de um século como um conjunto complexo de relações entre membros de um sistema social, conceito esse que também se aplica à rede social presente na internet. De acordo com o autor, a era moderna das redes sociais começou em 2002, com a rede social “Friendster”. De lá para cá, várias redes surgiram, a maioria com o conceito de um grupo de pessoas unidas através de um interesse comum, compartilhando informações e discutindo sobre os mais diversos assuntos.

A utilização de uma rede social para o Terceiro Setor possui dois objetivos, o primeiro é realizar um elo de ligação entre voluntário e organização, através da utilização da internet, convocando pessoas para abraçar uma determinada causa e envolve-las em oportunidade de voluntariado; o segundo é permitir que as pessoas e organizações interajam umas com as outras, realizando publicações, comentários, e curtidas. Em uma pesquisa realizada na Internet, foram localizados vários sites com o propósito de realizar essa ligação entre voluntário e organizações, mas a maioria não tem o apelo de uma Rede Social, conforme pode ser observado no QUADRO 1.

QUADRO 1 - PESQUISA DE REDES SOCIAIS DE VOLUNTARIADO

Site	Rede Social
<a href="https://www.atados.com.br/">https://www.atados.com.br/</a>	Não
<a href="https://www.soprodeamor.com.br">https://www.soprodeamor.com.br</a>	Não
<a href="http://institutoalachaster.org/conexoes/programa-voluntariado">http://institutoalachaster.org/conexoes/programa-voluntariado</a>	Não
<a href="https://euresolvo.org.br/">https://euresolvo.org.br/</a>	Não
<a href="http://voluntarios.com.br/">http://voluntarios.com.br/</a>	Não
<a href="https://www.acaovoluntaria.org.br/">https://www.acaovoluntaria.org.br/</a>	Não
<a href="http://www.cruzvermelha.org.br/pb/institucional/">http://www.cruzvermelha.org.br/pb/institucional/</a>	Não
<a href="http://www.redevoluntariado.com.br/">http://www.redevoluntariado.com.br/</a>	Não
<a href="https://www.institutoponte.org.br/contribua/seja-um-voluntario/">https://www.institutoponte.org.br/contribua/seja-um-voluntario/</a>	Não
<a href="https://www.fundacaoayni.org/">https://www.fundacaoayni.org/</a>	Não
<a href="http://www.e-solidario.com.br/index.xhtmll">http://www.e-solidario.com.br/index.xhtmll</a>	Não
<a href="https://www.napratica.org.br/">https://www.napratica.org.br/</a>	Não
<a href="https://www.unv.org/become-volunteer">https://www.unv.org/become-volunteer</a>	Não
<a href="https://www.givegab.com/">https://www.givegab.com/</a>	Sim

FONTE: Os Autores (2018).

O único com características de rede social, é o GiveGab, plataforma sem fins lucrativos, cujas funcionalidades se destacaram perante os demais. É uma plataforma bonita, intuitiva e sem erros de sistema aparentes. Possibilita cadastrar tanto o perfil de uma organização quanto o perfil de um voluntário, disponibiliza várias formas de buscar oportunidades de voluntariado, possibilita procurar pessoas e solicitar amizade, permite realizar publicações, compartilhamentos, curtidas e comentários,

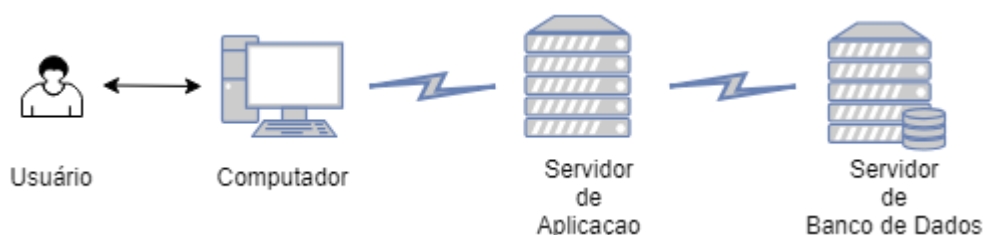
além de permitir também realizar doações em dinheiro. Para organizações, existem planos de assinatura, começando do básico até pacote empresarial.

## 2.2 ASPECTOS CONCEITUAIS DA TECNOLOGIA ENVOLVIDA

Através do uso da tecnologia, esse projeto foi desenvolvido com o intuito de facilitar a realização de trabalho voluntário, aproximando os envolvidos nessa atividade, que é realizada ao redor do mundo inteiro, por meio da criação de uma rede social específica para essa finalidade.

Para a criação da rede social, foram utilizados os principais recursos apresentados ao longo do curso, com foco em desenvolvimento de aplicações para a internet, prerrogativa para que uma rede social online funcione. A FIGURA 2 demonstra, de forma simplificada, um exemplo de estrutura de uma aplicação web.

FIGURA 2 - EXEMPLO DE ESTRUTURA



FONTE: Adaptada de JANDL JUNIOR (2015).

Pela FIGURA 2, é possível entender que, ao acessar o sistema através de um computador, o usuário executa o sistema que está hospedado em um servidor de aplicação e este, por sua vez, comunica-se com o servidor de banco de dados para buscar as informações armazenadas (JUNIOR, 2015). Existe a possibilidade de utilizar o mesmo servidor para instalar a aplicação e o banco de dados, no entanto, dependendo do porte do sistema, o recomendável é separá-los. Nas seções seguintes, apresenta-se cada uma das tecnologias utilizadas.

### 2.2.1 Sistema Gerenciador de Banco de Dados

Um banco de dados é um conjunto organizado de dados que são armazenados com o objetivo de gerar informação. Para realizar a manutenção desses dados, é

necessária a utilização de um Sistema de Gerenciamento de Banco de Dados (SGBD), pois é através dele que os dados serão incluídos, excluídos, consultados ou atualizados por um usuário ou por uma aplicação (RAMAKRISHANAN, 2008).

Os primeiros sistemas de gerenciamento de banco de dados começaram a surgir a partir da década de 1960 e de lá para cá não pararam de surgir diferentes modelos. O primeiro, chamado de Depósito de Dados Integrados, foi criado por Charles Bachman e foi a base do modelo de dados de rede. Em seguida, criado pela IBM, surgiu o IMS (Information Management System), que baseou o modelo de dados hierárquico. Já em 1970, apareceu o modelo de dados relacional que se destacou e é um dos mais utilizados até os dias de hoje, sendo o escolhido para o desenvolvimento de nossa aplicação. Outros modelos que surgiram foram o modelo orientado a objetos, o modelo objeto-relacional e mais atualmente o modelo NoSQL (RAMAKRISHANAN, 2008).

O modelo relacional possui esse nome devido ao fato de os registros serem armazenados em tabelas na forma de uma relação, como pode ser visto no exemplo (QUADRO 2):

QUADRO 2 - TABELA DE USUÁRIOS CADASTRADOS

Usuário	E-mail	Senha
1	usuario1@email.com	d9d8019ac
2	usuario2@exemplo.com	9acd91d80

FONTE: Os Autores (2018).

Do QUADRO 2, pode-se retirar alguns conceitos importantes, são eles:

- a) Cada linha representa um registro, também conhecido por tupla e identifica um usuário em nosso exemplo;
- b) Cada registro é composto por um ou mais atributos (colunas/campos). Esses atributos devem possuir um domínio, identificando que tipo de informação pode ser incluída em cada um deles (números inteiros, cadeia de caracteres, número reais, etc.);
- c) Entidade: é a representação de um objeto do mundo real, distinta de outros objetos, também conhecida como tabela, no nosso caso, é a tabela aluno;
- d) Relacionamento: Caso a tabela acima, possua algum vínculo com outra tabela do banco de dados, através de um dos seus atributos, o mesmo é denominado de relacionamento. No exemplo, pode-se dizer que a tabela de usuário tem relação

com a tabela endereço, que conterá os atributos relativos a essa entidade (endereço) de cada usuário cadastrado.

Além das tabelas, um sistema gerenciador de banco de dados possui outros objetos, como por exemplo, visões, procedimentos armazenados e gatilhos, entre outros, organizados dentro de um Schema cuja organização desses objetos e seus relacionamentos formam o projeto relacional de um banco de dados (RAMALHO, 2002).

### 2.2.2 Linguagens de Programação

As linguagens de programação foram criadas para que se transmita ao computador quais instruções devem ser realizadas. Pelo fato de o computador entender apenas instruções binárias, no nível mais próximo ao hardware, é necessária a utilização de uma linguagem em que se possa escrever comandos na língua natural, que serão convertidos em instruções compreensíveis pelo computador (KANAN, 1998).

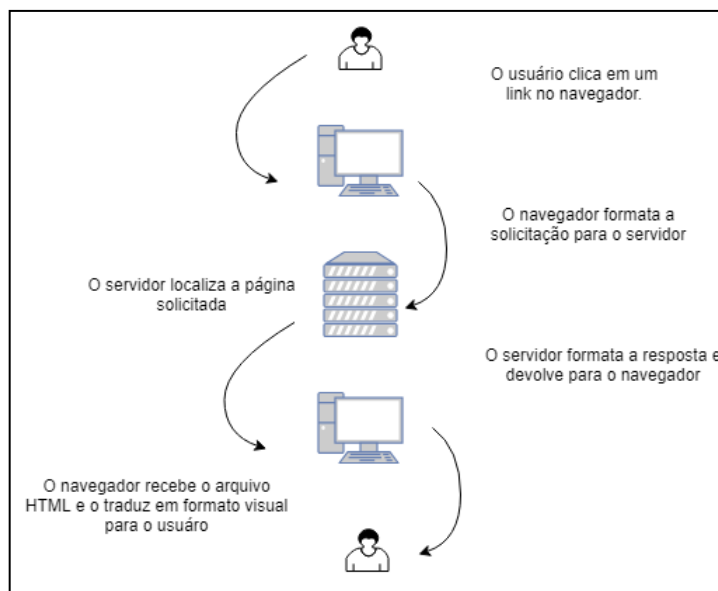
Nas subseções seguintes, apresentamos as linguagens de programação utilizadas para a elaboração do projeto, sendo elas: HTML, JavaScript, CSS, SQL e Java, linguagens essas estudadas ao longo do curso e que possibilitaram a realização do software.

#### 2.2.2.1 HTML

Segundo Savola (1997, p. XXII), existe um consenso de que a HTML, acrônimo de Hypertext Markup Language, seja considerada uma linguagem de programação, reforçado principalmente pela quantidade de programadores que atuam no mercado de trabalho da *World Wide Web*. É uma linguagem de marcação, composta por indicadores (elementos) que definem o texto e os objetos do documento, permitindo formatá-los e criar relacionamentos (conhecidos por links de hipertexto) entre eles. Os documentos criados com a linguagem são conhecidos como páginas *web* e sua utilização sem outra tecnologia limita-se a criação de página estáticas, que apresentam sempre o mesmo conteúdo, dependendo de uma pessoa para realizar a atualização das informações ali contidas. A FIGURA 3 demonstra a exibição em um

computador denominado cliente, de uma página HTML armazenada em um outro computador, denominado servidor web.

FIGURA 3 - CLIENTE X SERVIDOR WEB



FONTE: Adaptada de Basham, Bryan; Sierra, Kathy; Bates, Bert (2008).

Segundo Ramalho, (1999, p. 16), um programa HTML é composto por três partes básicas, a estrutura principal, o cabeçalho e o corpo do programa. O programa pode ser escrito em um editor de texto simples, porém existem editores HTML que ajudam a economizar tempo na sua construção. Dentro do arquivo, são digitados símbolos especiais, conhecidos por tags, que informam ao navegador como a informação deve ser exibida, além de redirecionar para outras páginas. As tags são identificadas por pares de sinais que definem seu início e fim, para exemplificar abaixo são elencados alguns exemplos:

- 1) `<b>casa</b>`: Ao interpretar a tag `<b>`, o navegador apresentará em negrito o que estiver escrito até encontrar a tag `</b>`, no caso a palavra casa.
- 2) `<a href = "www.google.com.br">Google</a>`: Ao interpretar a tag `<a href>`, o navegador criará um link na palavra Google, redirecionando o usuário para a referida página, caso o usuário clique nela.



### 2.2.2.2 JavaScript

Comumente as pessoas confundem a linguagem JavaScript com a linguagem Java, explicada na seção 2.2.2.5, entretanto tratam-se de duas linguagens de programação totalmente distintas.

Damiani (2006, p. 7) cita que foi criada em 1995 pela Netscape, com o objetivo de permitir a manipulação dinâmica de uma página HTML em seu navegador (Netscape), devido ao sucesso, sua concorrente, a Microsoft®, criou uma versão própria, chamada JScript. Posteriormente houve uma padronização da linguagem, para que funcionasse em qualquer navegador, realizada pelo ECMA (European Computer Manufactures Association), surgindo o padrão ECMAScript.

Também é uma linguagem orientada a objetos, possuindo os conceitos de encapsulamento, herança e polimorfismo. Porém, é baseada em protótipos, ou seja, não há necessidade de instanciar classes, tornando-a mais dinâmica. O propósito da linguagem é realizar ações na página HTML, mediante eventos disparados pelo usuário durante a navegação. O código pode ser colocado dentro do próprio arquivo HTML ou em arquivo à parte, apenas indicando na página sua localização, sendo essa última forma a mais recomendada por questões de estética e clareza. A FIGURA 4 exemplifica a forma de indicação em arquivo à parte.

FIGURA 4 - LINK ENTRE HTML E JAVASCRIPT

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Style Sheet Demo</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<script src="/js/voluntarius.js"></script>
</head>
```

FONTE: Os Autores (2018).

Segundo Kalbach (2009, p. 207), através da utilização da linguagem é possível combinar essa tecnologia para fazer com que as páginas web respondam de forma mais rápida, realizando troca de pequenas quantidades de dados com o servidor, evitando que a página inteira tenha que ser recarregada toda vez que o usuário realiza uma mudança. É o caso do AJAX, acrônimo de *Asynchronous JavaScript and XML*,

que combina a linguagem *Javascript* com o formato de arquivo XML, que é um tipo de arquivo texto onde as informações são separadas por *tags*.

### 2.2.2.3 CSS

Segundo Kalbach (2009, p. 206), o CSS, acrônimo de *Cascading Style Sheets*, possibilita definir o estilo para todo o site, incluindo cores, fontes e layout, por exemplo. Sua utilização traz como vantagem a possibilidade de separar a formatação do conteúdo da estrutura do documento HTML. Isso é útil pois pessoas com habilidades diferentes podem trabalhar na produção de uma página HTML focando naquilo que possuem melhor conhecimento.

Shafer (2003, p. 14) explica que essa definição pode ser feita de três maneiras em uma página HTML:

- 1) Definindo o estilo dentro de uma *tag* existente. Exemplo: `<h2 style = "color: red; text-transform: uppercase;"> Cabeçalho em fonte vermelha com letras maiúsculas</h2>`.
- 2) Incorporando as definições do estilo na seção cabeçalho, ainda dentro da página HTML, conforme demonstrado na FIGURA 5.

FIGURA 5- CSS INCORPORADO AO HTML

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Style Sheet Demo</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style type = "text/css">
  h1, h2 {
    color: green;
  }
  h3 {
    color: blue;
  }
</style>
</head>
...
```

FONTE: Os Autores (2018).

- 3) Em um arquivo totalmente separado da página HTML, definindo o link para o arquivo de estilos, conforme representado na FIGURA 6.

FIGURA 6 - LINK ENTRE HTML E CSS

```

<!DOCTYPE html>
<html>
<head>
<title>CSS Style Sheet Demo</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style type = "text/css">
    h1, h2 {
        color: green;
    }
    h3 {
        color: blue;
    }
</style>
</head>
...

```

FONTE: Os Autores (2018).

Shafer explica ainda que, a definição de estilos é composta por uma ou mais regras. Cada regra é composta por um seletor e propriedades (ou atributos). O seletor serve para definir em qual elemento da página HTML será aplicada e as propriedades servem para definir a aparência de todos os elementos que coincidirem com o seletor definido. Na FIGURA 6, por exemplo, percebe-se que em todas as *tags* de cabeçalho nível 1 e 2 (h1, h2), será aplicada a cor verde para o texto, enquanto as *tags* de cabeçalho de nível 3 (h3) possuirão o texto na cor azul.

#### 2.2.2.4 SQL

Segundo Souza (2004, p. 29-30), a linguagem SQL (Structured Query Language) é um conjunto de comandos que programas e usuários usam para ter acesso aos dados dentro de um banco de dados relacional. Sua origem é devida à publicação de um trabalho realizado por E. F. Codd, em 1970, intitulado *A Relational Model of Data for Large Shared Data Banks*. A linguagem foi adotada como padrão para Banco de Dados relacionais, pelo Instituto de padrões nacional americano (ANSI) e pela International Standards Organization (ISO) e é através dele que se realiza manutenções no banco de dados.

Ramalho (2002, p. 138) explica que os comandos SQL são divididos em duas categorias principais: DDL (Data Definition Language), ou linguagem de definição de dados, responsável por definir dados e objetos de um banco de dados; e DML (Data Manipulation Language), ou linguagem de manipulação de dados, responsável pela

realização de consultas e alterações das tabelas. Os principais comandos das duas categorias são apresentados no Quadro 3 com sua respectiva função.

QUADRO 3 - COMANDOS SQL

Comando	Categoria	Função
CREATE TABLE	DDL	Criar tabela no banco de dados.
ALTER TABLE	DDL	Alterar tabela no banco de dados.
DROP TABLE	DDL	Excluir tabela do banco de dados.
INSERT	DML	Inserir registro(s) em uma tabela.
DELETE	DML	Excluir registro(s) de uma tabela.
UPDATE	DML	Atualizar registro(s) de uma tabela.
SELECT	DML	Buscar registro(s) de uma tabela.

FONTE: Os Autores (2018).

#### 2.2.2.5 Java

A linguagem Java™ foi apresentada formalmente em maio de 1995 pela Sun Microsystems®, que deseja uma linguagem simples, portátil e fácil de programar. Seu nome inicial era Oak e começou a ser desenvolvida em 1993. Segundo Jandl Junior (2015, p. 17), poderia ser descrito da seguinte maneira:

O Java é uma linguagem de programação de propósito geral, concorrente, baseada em classes e orientada a objetos. Projetada para ser simples o bastante para que a maioria dos programadores se torne fluente na linguagem. Java tem relação com C e C++, porém é organizada de forma diferente, com vários aspectos dessas linguagens omitidos e algumas ideias de outras linguagens incluídas (JUNIOR, PETER JANDL, 2015, p. 17).

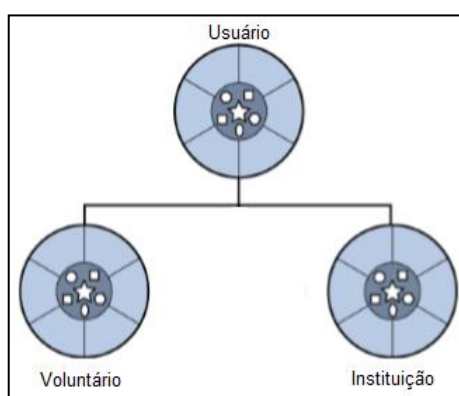
A partir de 2009, a Sun foi comprada pela Oracle® que atualmente organiza a linguagem em três segmentos:

- 1) Java ME (Java Micro Edition), destinado a dispositivos computacionais móveis como celulares, por exemplo;
- 2) Java SE (Java Standard Edition), destinado ao desenvolvimento de aplicações de pequeno e médio porte, contempla os elementos padrões da plataforma;
- 3) Java EE (Java Enterprise Edition), para o desenvolvimento de aplicações corporativas complexas, contendo mais recursos que a anterior. Recentemente, em setembro de 2018, a Eclipse Foundation assumiu uma versão voltada ambiente cloud, denominada Jakarta EE. A partir de

setembro de 2018, essa versão foi assumida pela Eclipse Foundation e distribuída pelo nome Jakarta EE.

Trata-se de uma linguagem orientada a objetos, diferente da linguagem C, por exemplo, que é uma linguagem estruturada. Isso significa que possui recursos de abstração, encapsulamento e hereditariedade. Ao desenvolver uma aplicação, criam-se classes para representar um objeto do mundo real e utiliza métodos para interagir com o objeto representado. Na FIGURA 7, representamos três classes criadas na rede social, com a explicação de alguns conceitos importantes:

FIGURA 7 - EXEMPLO DE CLASSES



FONTE: Adaptada de WANDRESEN; RAFAEL ROMUALDO (2017).

Da FIGURA 7, pode-se entender que um usuário no sistema é representado por uma classe. Os atributos na FIGURA 7 são representados pelos desenhos no círculo interior e poderiam ser, nesse caso, o código, o e-mail, o nome e a senha do usuário. Em volta desse círculo interior, estão os métodos, estruturas utilizadas para consultar ou realizar qualquer alteração nos atributos. Somente através dos métodos é que podemos realizar qualquer interação com os atributos, a isso chamamos de encapsulamento (JANDL JUNIOR, 2015).

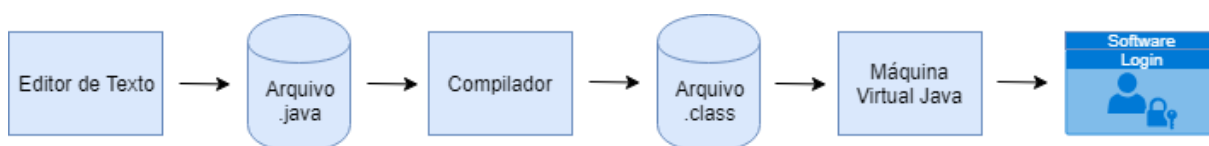
Outro conceito importante é o de hereditariedade, representado na FIGURA 7 pelas classes Voluntário e Instituição. Essas classes herdaram os atributos da classe usuário, mas possuem atributos específicos, como CPF e CNPJ, por exemplo, tornando-as tipos de usuários distintos (JANDL JUNIOR, 2015).

Existe o conceito de polimorfismo, que é a possibilidade de implementar os métodos de várias formas, no exemplo, um método na classe usuário pode ser

reescrito de forma diferente na classe voluntário e instituição, cada um atendendo a particularidade de seu objeto (JANDL JUNIOR, 2015).

Um último conceito importante dessa linguagem é que, segundo Jandl Junior (2015, p. 21-22), as aplicações desenvolvidas não geram um arquivo executável dependente de sistema operacional. A linguagem gera um arquivo com extensão “.class” com instruções denominadas “*bytecodes*”, que pode ser executado tanto em um ambiente Windows® quanto em um ambiente Linux, através de um aplicativo chamado JVM (*Java Virtual Machine*). A JVM precisa ser compatível com o Sistema Operacional. A FIGURA 8 ilustra a explicação.

FIGURA 8 - CÓDIGO-FONTE, BYTECODES E CÓDIGO NATIVO



FONTE: Adaptada de JANDL JUNIOR; PETER (2015).

A linguagem possui ainda componentes adicionais, que permitem a criação de aplicações web com páginas HTML dinâmicas, bem como a comunicação com sistemas gerenciadores de bancos de dados. Alguns desses recursos merecem destaque dada a sua importância na elaboração do projeto.

O primeiro componente a ser apresentado se chama *Servlets*. Segundo Furgeri (1996, p. 329), são módulos de software compilado (objetos), que ficam armazenados em um servidor Java, necessário para sua execução. Silberschatz (2006, p. 214) explica que são classes que recebem informações através de uma requisição, utilizam-na para algum tipo de processamento, e devolvem uma resposta contendo informações. Através delas, é possível controlar a sessão de um usuário, verificando se o mesmo ainda está conectado ou não.

Possuem ciclo de vida definido, controlado pelo servidor Web onde estão instaladas e caso não estejam instanciadas, o servidor invoca o método “*init ()*” para inicia-las. Uma vez iniciada, o servidor invoca o método “*service ()*”, que pode ser “*doGet ()*” ou “*doPost ()*”, com um objeto *request* e um objeto *response* como parâmetros, em um novo *thread*, isso se faz necessário para que o servidor possa processar várias solicitações ao mesmo tempo. As *Servlets* são finalizadas através do método “*destroy ()*”, que pode ser configurado no servidor para ser executado depois

de um determinado tempo, ou através de chamada explícita, diretamente no código de programação.

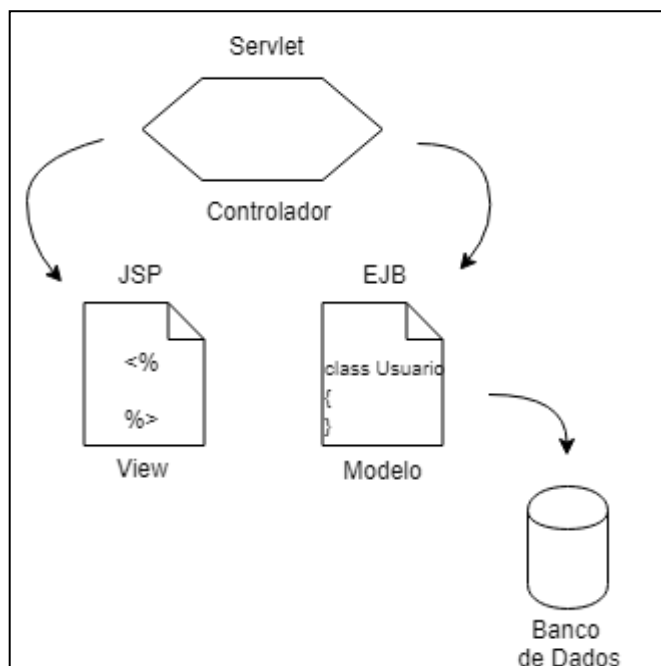
Outro componente importante, chamado *JSP (JavaServer Pages)*, baseia-se na tecnologia *Servlet* e tem como objetivo simplificar o desenvolvimento de conteúdos dinâmicos na Web. Na JSP é escrito código HTML junto com instruções que serão substituídas por conteúdo dinâmico, no momento de sua criação. Ao serem processadas pelo servidor de aplicação, o contêiner converte-as em *Servlets*. Entre as instruções, cabe destacar a *EL (Expression Language)*, que é uma forma de executar código *Java* sem escrever métodos na *JSP*, e a *JSTL (JSP Standard Tag Library)*, que é um apanhado das *tags* mais utilizadas e que facilitam a construção de ações como um *loop* ou um condicional. Também é possível escrever código *Java*, recurso conhecido como *Scriptlets*, embora essa prática não seja recomendada pois mistura a parte de negócio com a parte de visualização da página tornando o código confuso (BASHAM, 2008).

O Contêiner é mais um componente importante. Segundo Furgeri (1996, p. 355), é uma interface entre um componente e uma plataforma específica que possibilita definir ou restringir as ações de certos componentes. O Contêiner é responsável por instanciar a *Servlet*, criar um novo “thread” para atender a uma solicitação, chamar os métodos “*doGet ()*” ou “*doPost ()*”, levar a *request* e *response* para as *Servlets* e controlar o ciclo de vida delas (BASHAM, 2008).

O último, *Enterprise Java Beans (EJB)*, é conhecido como um componente de negócio, que atua ao lado do servidor e encapsula a lógica de negócio em uma aplicação, permitindo a um cliente (navegador) realizar operações definidas em métodos implícitos na aplicação. Utilizando esse componente, o programador se preocupa com a lógica do negócio a ser desenvolvido, sem se preocupar com detalhes das transações como segurança, conexões, desempenho do servidor, questões essas tratadas pelo contêiner do EJB (FURGERI, 1996).

A utilização dos componentes *JSP*, *EJB* e *Servlets* pode ser feita de uma forma padronizada, conhecida por MVC (*Model – View – Controller*). A FIGURA 9 ilustra essa arquitetura.

FIGURA 9 - PADRÃO MVC



FONTE: Adaptada de Basham, Bryan; Sierra, Kathy; Bates, Bert (2008).

O padrão de Façade é design de software comumente usado em programação orientada a objetos. Analogamente a uma fachada na arquitetura, uma Façade é um objeto que serve como uma interface frontal que mascara o código subjacente ou estrutural mais complexo. Uma Façade pode:

1. melhorar a legibilidade e a usabilidade de uma biblioteca de software, mascarando a interação com componentes mais complexos por trás de uma única (e muitas vezes simplificada) API;
2. fornecer uma interface específica do contexto para uma funcionalidade mais genérica (completa com validação de entrada específica do contexto);
3. fornecer uma interface específica do contexto para uma funcionalidade mais genérica (completa com validação de entrada específica do contexto);
4. servir como um ponto de partida para um refator mais amplo de sistemas monolíticos ou fortemente acoplados em favor de um código mais fracamente acoplado.

Costuma-se usar o padrão de design de Façade quando um sistema é muito complexo ou difícil de entender porque o sistema possui um grande número de classes interdependentes ou porque seu código-fonte não está disponível. Esse padrão esconde as complexidades do sistema maior e fornece uma interface mais simples



para o cliente. Geralmente, envolve uma única classe de wrapper que contém um conjunto de membros requeridos pelo cliente. Esses membros acessam o sistema em nome do cliente de Façade e ocultam os detalhes da implementação.

O Padrão de Objeto de Acesso a Dados ou padrão DAO é usado para separar APIs ou operações de acesso a dados de baixo nível de serviços de negócios de alto nível. A seguir estão os participantes no Padrão de Objeto de Acesso a Dados:

1. Data Access Object Interface - Essa interface define as operações padrão a serem executadas em um objeto de modelo;
2. Classe concreta do Objeto de Acesso a Dados - Esta classe implementa a interface acima. Essa classe é responsável por obter dados de uma fonte de dados que pode ser database / xml ou qualquer outro mecanismo de armazenamento;
3. Objeto de modelo ou objeto de valor - Este objeto contém métodos get / set para armazenar dados recuperados usando a classe DAO.

### 2.2.3 Modelagem - UML

Ao desenvolver um software orientado a objetos, é importante realizar a representação gráfica de sua modelagem antes de iniciar o trabalho de desenvolvimento em si. Em sua obra, Booch (2012, p. VII) define a UML como uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software.

Sua origem tem início em meados dos anos 90, quando Grady Booch, Ivar Jacobson e James Rumbaugh, representantes dos métodos Booch, OOSE (*Object Oriented Software Engineering*) e OMT (*Object Modeling Technique*) respectivamente, se reuniram e começaram a utilizar ideias provenientes desses três métodos, os quais mais se destacavam na época, criando a *Unified Modeling Language* (BOOCH, 2012).

Segundo Booch (2012), ao realizar a unificação dos métodos, tinham como três objetivos principais:

- 1) Fazer a modelagem de sistemas, do conceito ao artefato executável, com a utilização de técnicas orientadas a objetos;

- 2) Incluir questões de escala, inerentes a sistemas complexos e de missão crítica;
- 3) Criar uma linguagem de modelagem a ser utilizada por seres humanos e por máquinas.

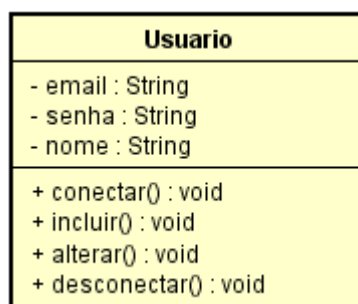
A linguagem é composta por itens, relacionamentos e diagramas, explicados individualmente, sem a intenção de esgotar o assunto, nas seções seguintes.

### 2.2.3.1 Itens da UML

Os itens são definidos como peça fundamental da UML e subdividem-se em itens estruturais, itens comportamentais, itens de agrupamento e itens anotacionais (BOOCH, 2012).

Os itens estruturais correspondem às partes mais estáticas do modelo e representam elementos conceituais ou físicos. Destacam-se as classes e os casos de uso como elementos principais utilizados nesse projeto. Conforme definido por Booch (2012, p. 19), classes são descrições de conjuntos de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica. A FIGURA 10 exemplifica a forma gráfica de uma classe.

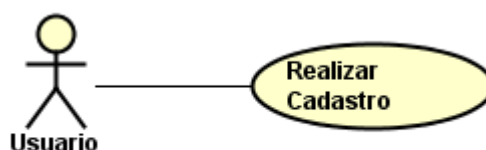
FIGURA 10 - REPRESENTAÇÃO GRÁFICA: CLASSE USUÁRIO



FONTE: Os Autores (2018).

A classe é composta por um título (Usuário), atributos (email, senha e nome) e operações que podem ser realizadas ao instanciar (“criar”) um objeto do tipo usuário. Já o caso de uso, representado na FIGURA 11, é conceituado por Booch (2012, p. 21) como a descrição de sequências de ações realizadas pelo sistema, que proporciona resultados observáveis de valor para um determinado ator.

FIGURA 11 - REPRESENTAÇÃO GRÁFICA: CASO DE USO REALIZAR CADASTRO



FONTE: Os Autores (2018).

Percebe-se na FIGURA 11 que o Caso de Uso é representado por uma elipse, com um nome único na modelagem, geralmente descrito com uma expressão verbal ativa. Ligado ao Caso de Uso, aparece a figura do Ator, definido por Booch (2012, p. 250) como aquele que representa um papel desempenhado por um ser humano, um dispositivo de hardware ou inclusive outro sistema. Segundo Booch (2012, p. 251), é importante observar que o um caso de uso descreve *o que* um sistema faz, mas não especifica *como* é feita a implementação, sendo importante manter de forma clara a separação entre a visão interna e externa. A especificação do caso de uso demonstra o seu comportamento através da descrição do fluxo de eventos principal, alternativo, regras de negócio e interações com atores, de forma clara, possibilitando que alguém de fora possa compreendê-lo facilmente.

Os itens comportamentais são compostos por interações, máquinas de estado e atividades, referem-se à parte dinâmica dos modelos de UML e representam o comportamento dos objetos e a troca de informações entre si. Já os itens de agrupamento servem apenas para organizar a parte de codificação do sistema, são puramente conceituais, como se fossem pastas do sistema operacional, onde os demais itens (estruturais e comportamentais, principalmente) podem ser armazenados. Por último, os itens anotacionais servem para realizar anotações, comentários, esclarecimentos ou observações sobre qualquer elemento do modelo (BOOCH; 2012). Na FIGURA 12, é representado um exemplo desses elementos.

FIGURA 12 - ITENS COMPORTAMENTAIS, DE AGRUPAMENTO E ANOTACIONAIS



FONTE: Adaptada de BOOCH; GRADY (2012).

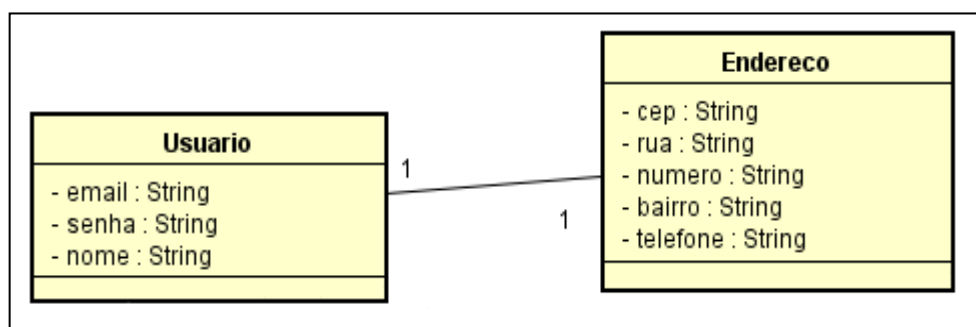
Cabe destacar desse grupo, devido à sua utilização no projeto, os conceitos que envolvem uma máquina de estado. Segundo Booch (2012, p. 24), é um comportamento que especifica as sequências de estados pelas quais objetos ou interações passam durante sua existência em resposta a eventos, bem como suas respostas a esses eventos. Abrange os seguintes elementos: estados, transições (o fluxo de um estado a outro), eventos (itens que disparam uma transição) e atividades (as respostas às transições).

### 2.2.3.2 Relacionamentos na UML

Conforme Booch (2012), os quatro tipos de relacionamento, que tem por função criar uma conexão entre os itens da modelagem, são a dependência, a associação, a generalização e a realização. Desses, destacamos a associação e a generalização por serem mais utilizados.

A associação é utilizada para representar o relacionamento entre as classes e é representada graficamente através de uma linha sólida, conectando a mesma classe ou classes diferentes. A ideia é transmitir a possibilidade de navegar do objeto de uma classe até o objeto da outra classe (BOOCH, 2012). A FIGURA 13 exemplifica a associação entre a classe usuário e a classe endereço.

FIGURA 13 - ASSOCIAÇÃO DE CLASSES

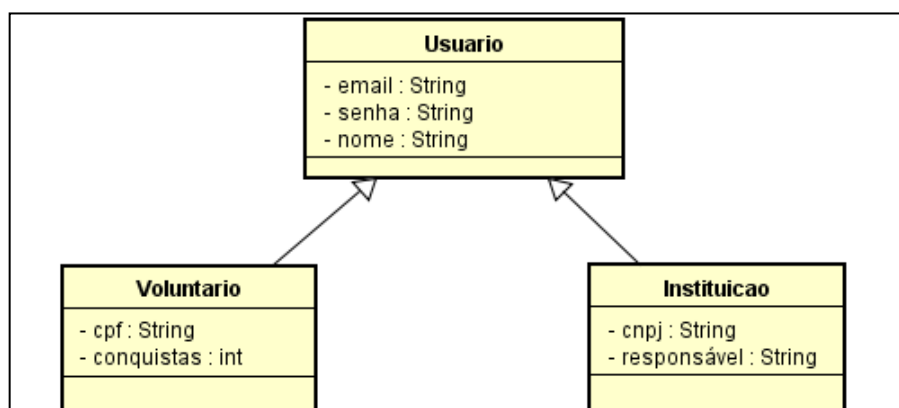


FONTE: Os Autores (2018).

Nela, é possível perceber a relação entre a classe usuário e a classe endereço, representada pelo traço e pelos números, chamados de multiplicidade, indicando que para cada usuário existente no sistema haverá um endereço e que cada endereço pertencerá a um único usuário. Além da associação simples, podem haver dois tipos especiais, denominados agregação e composição. O primeiro demonstra uma relação onde o elemento associado corresponde a uma parte do elemento principal, como por exemplo, o relacionamento entre a classe time e a classe atleta, cuja representação gráfica é realizada por um traço com um losango branco na ponta. O segundo, composição, é um tipo especial de agregação onde as partes pertencem somente a um determinado todo e sua representação gráfica é feita através de um traço com um losango preto na ponta (BOOCH, 2012).

A outra forma de relacionamento, generalização, é utilizado quando se deseja representar que uma classe é geral e a outra é especializada, podendo essa herdar elementos daquela. Representa-se graficamente por uma linha sólida com uma seta em branco apontando para a classe geral, também conhecida como classe mãe. Na FIGURA 14, é possível compreender melhor essa relação.

FIGURA 14 - GENERALIZAÇÃO



FONTE: Os Autores (2018).

Percebe-se pela FIGURA 14 que tanto Voluntário quanto Instituição são classes especializadas, que herdam algumas características da classe usuário, como por exemplo, o atributo nome.

### 2.2.3.3 Diagramas

Segundo Booch (2012), um diagrama é a apresentação gráfica de um conjunto de elementos, geralmente representadas como gráficos de vértices (itens) e arcos (relacionamentos), constituindo uma projeção do sistema modelado. Ao todo, são 13 diagramas na modelagem UML, cada qual com o seu objetivo. São eles:

- 1) Diagrama de classes;
- 2) Diagrama de objetos;
- 3) Diagrama de componentes;
- 4) Diagrama de estruturas compostas;
- 5) Diagrama de casos de uso;
- 6) Diagrama de sequências;
- 7) Diagrama de comunicações;
- 8) Diagrama de estados;
- 9) Diagrama de atividades;
- 10) Diagrama de implantação;
- 11) Diagrama de pacote;
- 12) Diagrama de temporização;

### 13) Diagrama de visão geral da interação.

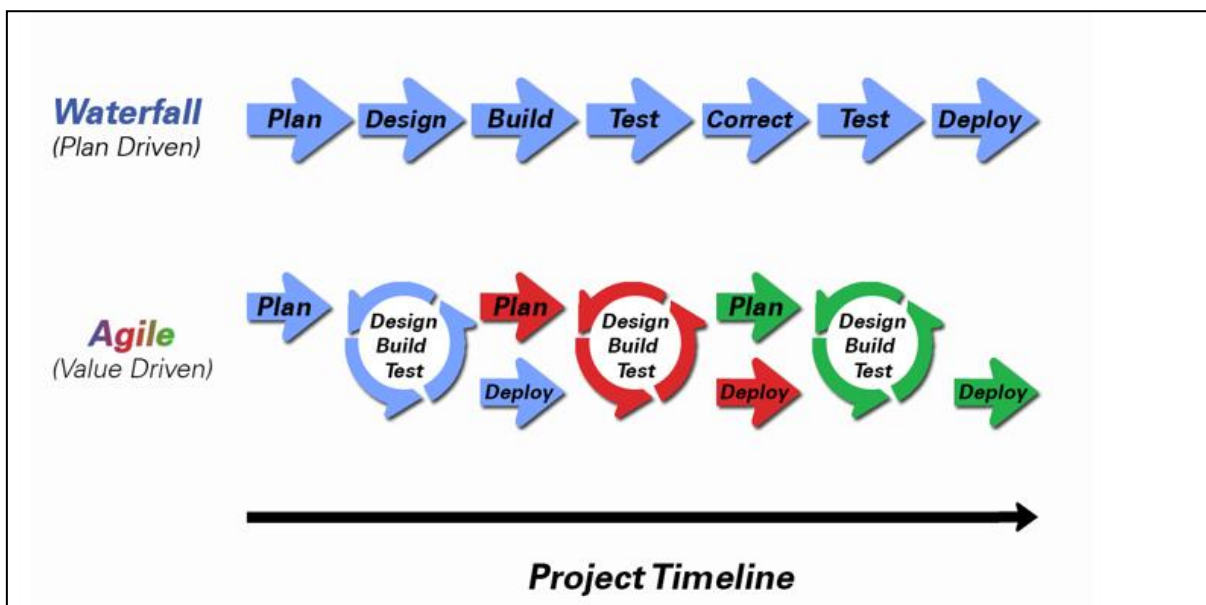
Desses, destacam-se o diagrama de classes, o diagrama de casos de uso, o diagrama de sequências e o diagrama de estados, todos utilizados na elaboração do projeto. Segundo Booch (2012), o diagrama de classes demonstra o conjunto das classes bem como seus relacionamentos, demonstrando uma visão de forma estática do sistema. Já o diagrama de casos de uso exibe uma visão geral de todas as funções do sistema, mostrando a interação entre seus elementos internos e externos. O diagrama de sequências, demonstra uma visão dinâmica do sistema, com as mensagens (informações) que podem ser trocadas entre seus elementos, demonstrando uma ordenação temporal. Por último, o diagrama de estados, representa uma máquina de estados, composta por estados, transições, eventos e atividades, cujo propósito é dar uma visão dinâmica de um objeto.

#### 2.2.4 Gerenciamento de Projetos

Assim como qualquer tipo de projeto, o desenvolvimento de software é um projeto que precisa ser gerenciado, de modo que ao seu término, tanto o cliente quanto o fornecedor da solução tenham retorno nos investimentos realizados, sejam eles de tempo, dinheiro ou qualquer outro recurso.

Segundo Martins (2007), a frequência com que projetos de software fracassaram, deu origem a diversas metodologias para gerenciamentos de projetos na área, seja no âmbito de desenvolvimento ou no de implantação de sistemas (ERP, por exemplo). Existe uma classificação que divide as metodologias criadas entre clássicas e ágeis. Na primeira, é realizado todo o planejamento antes de iniciar a execução, enquanto na segunda, inicia-se o projeto com um planejamento parcial e ao longo do processo de desenvolvimento, geralmente realizado em ciclos, novos planejamentos são realizados, incrementando o software e realizando novas entregas. A FIGURA 15 exemplifica a diferença entre uma metodologia e outra.

FIGURA 15 - METODOLOGIA CLÁSSICA X METODOLOGIA ÁGIL



FONTE: CAMARGO; ROBSON (2018).

A metodologia ágil surgiu em 2001, quando um grupo de dezessete pessoas criou um manifesto para o desenvolvimento ágil, contendo 12 princípios a serem seguidos, são eles:

- 1) Nossa maior prioridade é satisfazer o cliente através da entrega antecipada e contínua de software valioso;
- 2) Mudanças nos requisitos são bem-vindas, mesmo no final do desenvolvimento. Os processos ágeis aproveitam as mudanças para a vantagem competitiva do cliente;
- 3) Entregue software funcionando com frequência, de algumas semanas a alguns meses, com preferência pela menor escala de tempo;
- 4) As áreas de negócio e técnica (desenvolvedores) devem trabalhar juntos diariamente durante todo o projeto;
- 5) Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte de que precisam e confie neles para realizar o trabalho;
- 6) O método mais eficiente e eficaz de transmitir informações para e dentro de uma equipe de desenvolvimento é a conversa face a face;
- 7) Software funcionando é a principal medida de progresso;



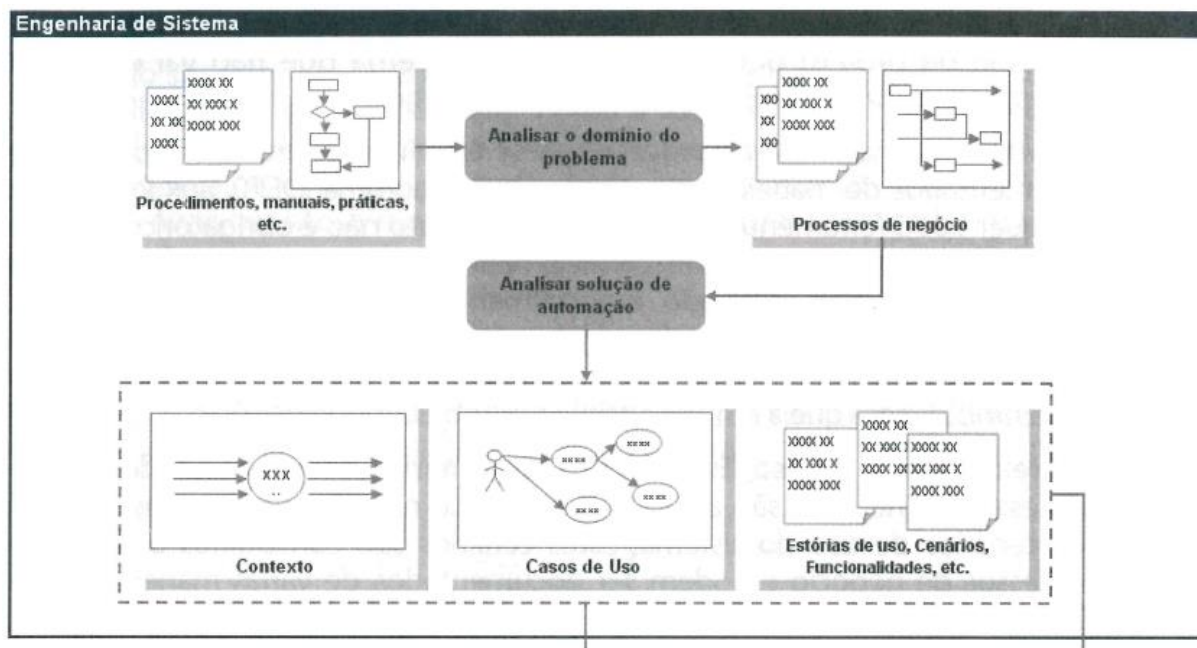
- 8) Processos ágeis promovem o desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente;
- 9) A atenção contínua à excelência técnica e ao bom design aumenta a agilidade;
- 10) Simplicidade - a arte de maximizar a quantidade de trabalho não realizado - é essencial;
- 11) As melhores arquiteturas, requisitos e projetos emergem de equipes auto organizadas;
- 12) Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz, depois ajusta e ajusta seu comportamento de acordo.

Segundo Martins (2007), esses valores e princípios indicam por que se deve agir de determinada maneira perante maneira nas variadas situações que aparecem durante o projeto, além de práticas que definem a forma de agir. *Scrum*, *Extreme Programming* (XP), *Kanban*, entre outros, são alguns exemplos de métodos ágeis.

Independente da abordagem, o projeto de desenvolvimento de software passa pelas seguintes etapas: engenharia de sistema, análise de requisitos, projeto técnico, construção e validação (MARTINS, 2007).

Martins (2007, p. 5) define a engenharia de sistema como o estudo do domínio do problema, dos processos de negócio, das necessidades e desejos dos clientes, das restrições de prazo e custo, e das restrições pertinentes à tecnologia e ao ambiente, conforme demonstrado na FIGURA 16.

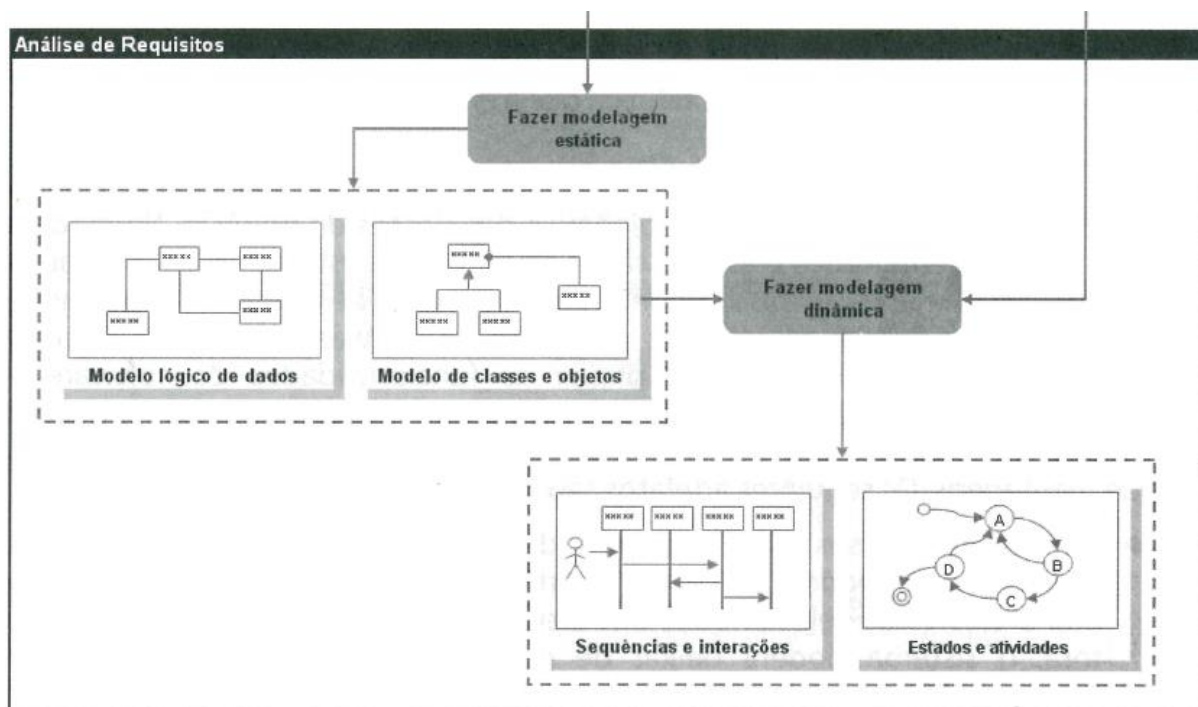
FIGURA 16 - ENGENHARIA DE SISTEMA



FONTE: MARTINS; JOSÉ (2007).

Segundo Martins (2007, p. 6), a análise de requisitos consiste na definição dos elementos básicos do software, fazendo a modelagem estática (diagrama de classes e entidades do banco de dados) e a modelagem dinâmica (diagrama de sequência e de estados), como pode ser visualizado na FIGURA 17.

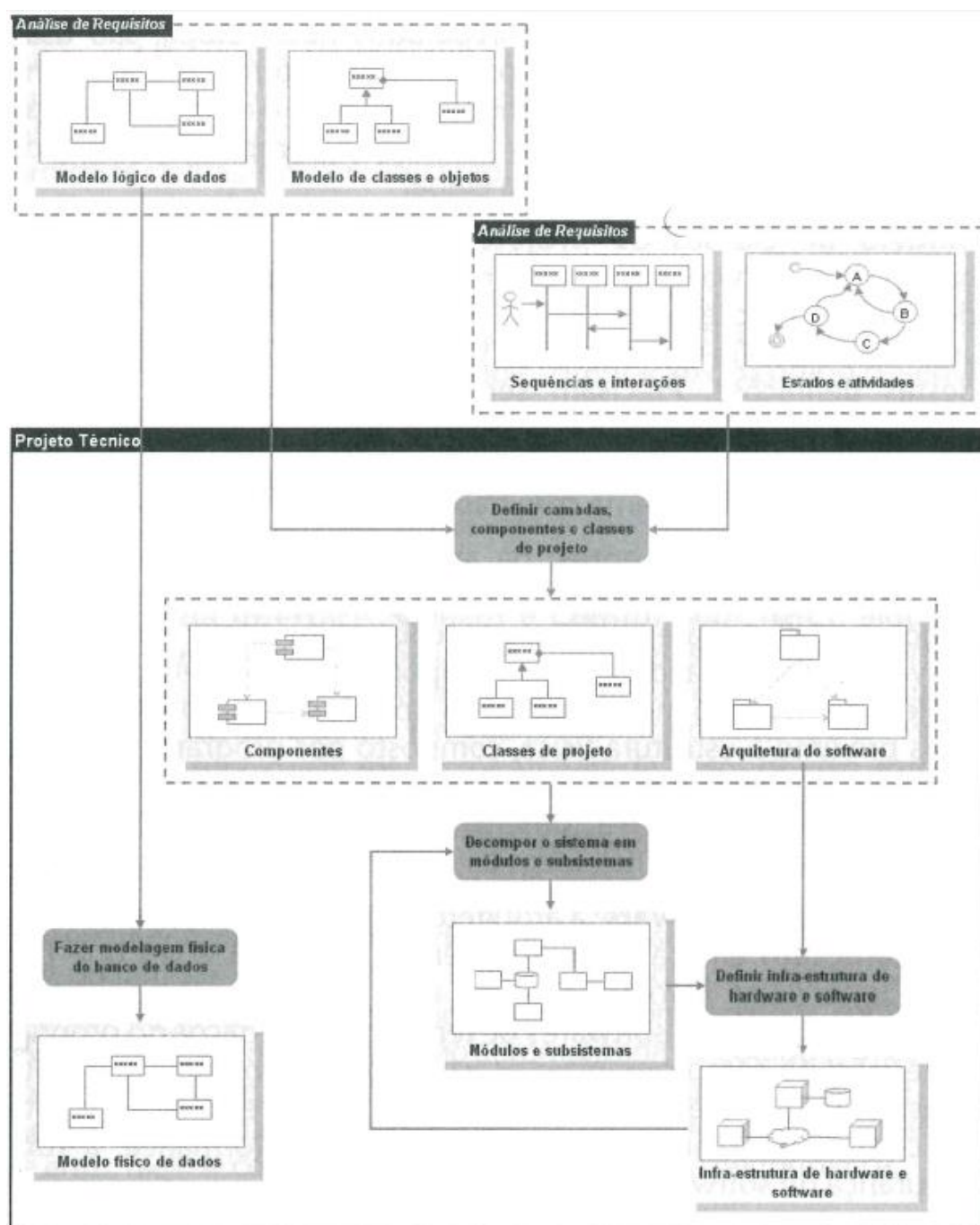
FIGURA 17 - ANÁLISE DE REQUISITOS



FONTE: MARTINS; JOSÉ (2007).

Para Martins (2007, p. 8), o projeto técnico possui a identificação das classes de projeto que possuem correspondência com o banco de dados e a interface com o usuário, dentre outras, além da decomposição do sistema em componentes e suas interfaces, mais os módulos de sistema, a forma como interagem, e na especificação do ambiente onde o software vai operar. Na FIGURA 18, é demonstrada essa etapa.

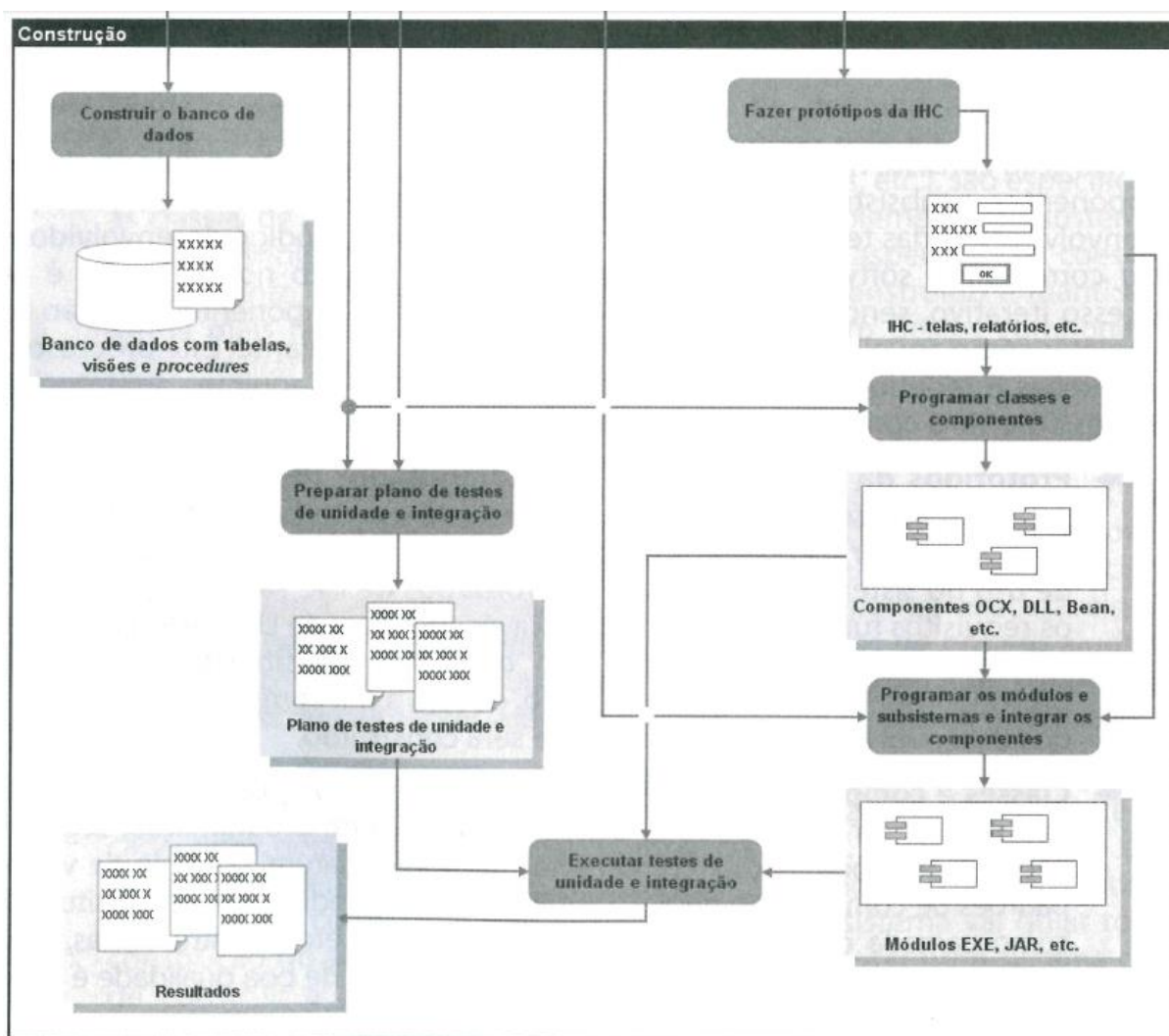
FIGURA 18 - PROJETO TÉCNICO



FONTE: MARTINS; JOSÉ (2007).

Na etapa de construção, Martins (2007, p. 9) explica que consiste no desenvolvimento em si, ou seja, escrever o programa na linguagem escolhida, gerando código-fonte, classes, entidades no banco de dados, telas e testes, conforme pode ser visualizado na FIGURA 19.

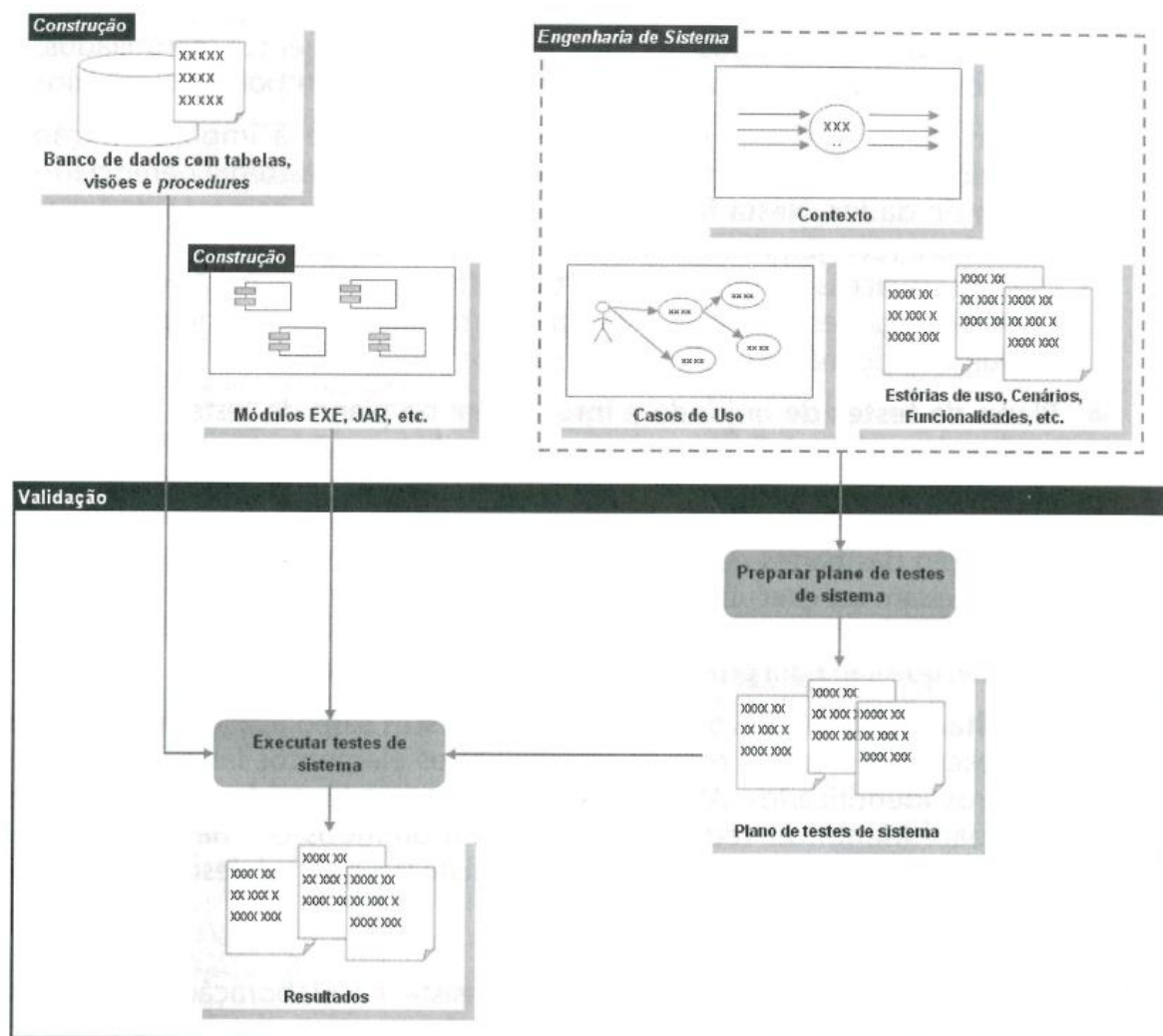
FIGURA 19 - CONSTRUÇÃO



FONTE: MARTINS; JOSÉ (2007).

Por último, na etapa de validação, segundo Martins (2007, p. 11), são feitos os roteiros e rotinas de teste, com o objetivo de validar o funcionamento do software e a homologação por parte do cliente, demonstrado na FIGURA 20.

FIGURA 20 - VALIDAÇÃO



FONTE: MARTINS; JOSÉ (2007).

Todas essas etapas podem ser executadas com maior ou menor grau de formalismo, na metodologia ágil, essas etapas se repetem em cada ciclo realizado, como se fossem miniprojetos, até sua conclusão final (MARTINS, 2007).

Existem três ferramentas, entre outras, que tem sua importância no gerenciamento de projetos de software e cabe aqui destacar devido à sua utilização na elaboração do projeto. A primeira delas, conhecida por EAP (Estrutura Analítica do Projeto), tradução em português de WBS (*Work Breakdown Structure*), a segunda, chamada de Diagrama de *Gantt* e a terceira, um repositório de código fonte, disponível na internet, conhecido por GIT.

Segundo Martins (2007), uma EAP consiste em montar um checklist identificando todas as partes de um projeto e suas respectivas tarefas. Sua vantagem

é facilitar o gerenciamento, subdividindo as tarefas em partes menores de modo que se consiga agendar, estimar custos, monitorar e controlar o trabalho planejado. É composta por dois tipos de elementos, os pacotes de trabalho, que correspondem a uma atividade realizada por um indivíduo ou equipe; e a tarefa resumo, que tem por finalidade agrupar esses pacotes de trabalho.

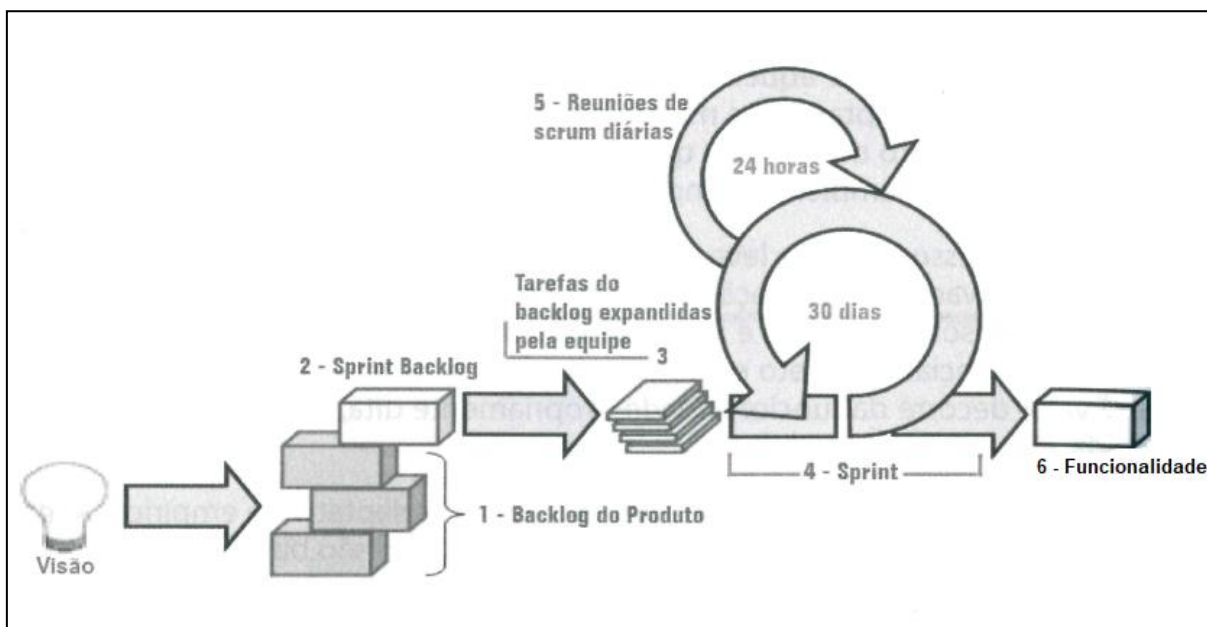
Vargas (2009, p. 174) explica que, o Diagrama de Gantt foi criado por *Henry Gantt* no início do século, com o objetivo de atender a fins militares e estratégicos. O Diagrama mostra a sequência das atividades com suas respectivas datas de execução demonstradas em forma de barras horizontais em um calendário. No gráfico, também são exibidos os recursos que estão alocados em cada atividade, bem como o esforço necessário em alguma escala de tempo, além do percentual de andamento de cada atividade. É uma ferramenta utilizada principalmente pelo gerente de projeto como forma de acompanhar o projeto e perceber alguma situação que comprometa a data de entrega.

#### 2.2.4.1 Scrum

O termo *Scrum* foi formalizado em 1995, por Jeff Sutherland e Ken Schwaber, quando trabalharam juntos na *Easel Corporation*. É utilizado no *Rugby*, no momento em que os jogadores se reúnem em círculo para realizar a próxima jogada e sua aplicação no desenvolvimento de software significa que o projeto deve ser conduzido em pequenos ciclos, com olho no objetivo final que é ganhar o jogo, ou seja, entregar o software funcionando. É um processo leve no gerenciamento e controle de projetos de desenvolvimento de software, por ser uma metodologia ágil, trabalha com a filosofia iterativa e incremental. (MARTINS, 2007).

Na FIGURA 21, é possível ver de forma resumida todo o processo do *Scrum*, que começa com uma visão, contendo uma ideia básica e que evolui à medida que o projeto começa a ser desenvolvido. Nessa ideia básica, é montada uma lista com os itens a serem desenvolvidos e que tornem a visão uma realidade, lista essa chamada de *Backlog* de Produto. Os itens constantes nessa lista são desenvolvidos em ciclos, chamados de *Sprint*, baseados em um documento denominando *Backlog* do *Sprint*, entregados as funcionalidades até o término do projeto.

FIGURA 21 - CICLO DO SCRUM



FONTE: MARTINS; JOSÉ (2007).

Martins (2007, p. 255) explica que na metodologia existem três papéis: o dono do produto, responsável por representar os interesses das pessoas que apostaram no projeto, podendo ser um gerente de projeto, e responsável por conseguir a verba, definir os requisitos gerais, os objetivos de ROI e o plano de *releases*; o Mestre *Scrum*, responsável por forçar os valores e as práticas do Scrum, remover obstáculos, ensinar o processo, implementar a metodologia e garantir que todos a sigam, é um líder e não se confunde com o gerente do projeto; e por último a equipe de projeto, responsável pelo desenvolvimento em si, composta por profissionais de diferentes especialidades, incluindo desenvolvedores e participantes externos de outras áreas (marketing, vendas, até mesmo clientes). Martins (2007, p. 256) explica que o *Scrum* possui uma estrutura composta por três fases: “*pré-game*”, “*game*” e “*pós-game*”, explicadas nas seções seguintes.

#### 2.2.4.1.1 Pré-Game

A primeira fase, chamada de “*pré-game*”, é composta por duas etapas: planejamento e arquitetura.

Na etapa de planejamento, onde ocorre o desenvolvimento do *Backlog*, a definição da equipe, avaliação de riscos e ações de controle, avaliação ou seleção



das ferramentas de desenvolvimento e infraestrutura, seleção do primeiro release, estimativa do custo do release (salário, matérias de consumo, treinamento, etc.) e obtenção de aprovação da direção para o projeto e para o orçamento. A FIGURA 22 demonstra uma lista de *backlog*.

FIGURA 22 - EXEMPLO DE BACKLOG

Backlog de Produto									
Super Monitor								Atualização	13/jan/05
								IED real diário	40%
Iter#	Id	Assunto	Item	Prior.	Resp.	IED	Risco	Dias estimados	Dias pendentes
1	15	Files/backup	Mudar arquitetura para trabalhar com filas de chamadas para processamento	1	AG	4	A	6.7	4.4
1	18	Watchdog	Criar mecanismo de watchdog e sistema de alarmes por e-mail	1	AG	4	M	6.7	5.9
1	19	Web	Migrar telas para ASP + IIS	1	AC	4	B	6.7	4.4
1	3	Segurança	Logar as pesquisas de gravação feitas pelos usuários e as reproduções	2	AC	0.5	B	0.8	3.1
1	8	Playback	Fazer reprodução através de playlist	2	AG	2	M	3.3	3.3
1	13	Segurança	Criptografar DVD com as gravações	2	AC	2	M	3.3	3.3
1	9	Playback	Enviar gravação por e-mail	3	AC	1	B	1.7	1.7
	4	Relatório	Relatório de auditoria - chamadas reproduzidas	2		0.5	B	0.8	?
	5	Relatório	Relatório de auditoria - acessos ao sistema	2		0.5	B	0.8	?
	6	Relatório	Relatório de auditoria - regras criadas	2		0.5	B	0.8	?
	7	Relatório	Relatório de auditoria - reproduções	?	2	0.5	B	0.8	?
	10	Relatório	Reproduzir gravação num ramal	2		5	A	8.3	?
	14	Regras	Criar regras de chamadas que não podem ser gravadas	2		3	M	5	?
	1	Segurança	Limitar acesso dos usuários por tela do sistema	2		2	M	3.3	?
	2	Segurança	Limitar quais gravações cada usuário pode ouvir, dependendo de quem foi gravado	2		1	M	1.7	?
	16	Gravação	Fazer gravação seletiva de chamadas	2		10	A	16.7	?
	11	Contr. Qualidade	Registrar comentários nas gravações, junto com nota para controle de qualidade	3		2	B	3.3	?
	12	Contr. Qualidade	Fazer relatório de controle de qualidade	3		1	B	1.7	?

FONTE: Adaptado de MARTINS, JOSÉ CARLOS CORDEIRO (2007).

Na elaboração do *backlog*, a coluna risco tem por função representar a certeza quanto à estimativa de tempo informada na coluna IED (*Ideal Engineering Day*), ou “Dia Ideal de Trabalho”, cuja definição seria aquele dia em que um desenvolvedor consegue dedicar 100% do tempo à implementação, sem interrupções. A partir disso, é feito um ajuste para o cálculo de dias estimados de trabalho em cada atividade, através da fórmula:  $IED / (1 - \% \text{ IED Real Diário})$ . O IED Real Diário é um percentual que representa a porcentagem de tempo que o desenvolvedor vai conseguir se dedicar ao projeto, levando em conta as interrupções normais do dia a dia (MARTINS, 2007).

Na etapa de arquitetura, os itens do *backlog* são projetados com vistas à implementação, realizando uma análise de domínio, um refinamento da arquitetura que suportará o contexto e os requisitos do sistema, a identificação de problemas que

possam vir a existir, a revisão do *backlog* para ajustes necessários e uma reunião de revisão de design, com o intuito de discutir a abordagem e possíveis mudanças necessárias em cada item do *backlog*.

#### 2.2.4.1.2 *Game*

A segunda fase, conhecida por *game*, é composta por um ou mais *sprints*, e seu objetivo é o desenvolvimento das funcionalidades em si, definidas no *backlog* do *sprint*. Martins (2007, 261) explica que o *Sprint* é um conjunto de atividades de desenvolvimento conduzidas num período de tempo predefinido, chamado de *time box*, com duração variando entre uma a quatro semanas. Cada *sprint* é composto por uma ou mais equipes realizando as atividades de desenvolvimento (análise, projeto técnico, programação, testes e documentação), empacotamento (criação dos arquivos executáveis para distribuição), revisão (discutir e revisar o progresso, resolver problemas e adicionar novos itens ao *Backlog*) e ajustes (com base nas informações coletadas na reunião de revisão, replanejando quando necessário).

Os *sprints* iniciam com uma reunião de planejamento, em que o Dono do Produto e a equipe decidem o que será feito na próxima iteração, cujo tempo de duração não deve durar mais que 8 horas e é dividida em duas partes: na primeira, o Dono do Produto apresenta e descreve os itens prioritários, esclarecendo eventuais dúvidas e questionamentos realizados pela equipe, que seleciona os itens a serem desenvolvidos na interação; enquanto na segunda parte, a equipe realiza o planejamento do *sprint* em si, criando o *backlog* do *sprint*. Durante o andamento de cada *sprint*, são realizadas reuniões diárias matinais com a participação do Mestre Scrum e todos os integrantes da equipe tendo como propósito sincronizar o trabalho diário, podendo outras pessoas participar apenas com ouvintes, para que não haja interferência (MARTINS, 2007).

Ao término de cada *sprint*, ocorrem mais duas reuniões: a primeira, reunião de revisão do *sprint*, onde é apresentado o trabalho realizado para o Dono do Produto e demais *stakeholder* interessados, com a intenção de obter *feedback* e determinar os objetivos da próxima iteração; e a segunda, reunião de retrospectiva do *Sprint*, com participação do Mestre Scrum e equipe, com a finalidade de fazer uma revisão no processo de trabalho e assim melhorar o desempenho na próxima iteração (MARTINS, 2007).

#### 2.2.4.1.3 Pós-Game

Na última fase, conhecida como pós-*game*, Martins (2007, p. 259) explica que após correr o fechamento do *release*, o produto é preparado para distribuição, realizando-se as atividades integração, testes adicionais documentação de usuário, preparação de material para treinamento e realização de marketing, entre outras.

### 3 MATERIAIS E MÉTODOS

No presente capítulo, é feita uma explanação da utilização do *Scrum* e as adaptações necessárias pelo fato de desenvolver o software em um ambiente acadêmico.

Além da aplicação metodologia, são apresentadas todas as ferramentas utilizadas para realizar o presente trabalho, como o hardware utilizado e os softwares de desenvolvimento, controle de projeto e edição de texto.

#### 3.1 APLICAÇÃO DO SCRUM E ADAPTAÇÕES

A justificativa para a utilização desse tipo de metodologia, já conceituada anteriormente, reside principalmente na realização de entregas semanais do software, apresentadas ao orientador, professor Razer Montañó, com o intuito de não só validar a funcionalidade desenvolvida, mas também de direcionar a sequência do desenvolvimento bem como esclarecer dúvidas técnicas que surgem durante o desenvolvimento de um software.

O papel de Dono do Projeto e Mestre *Scrum* foram desempenhados pelo orientador, que realizava as reuniões de planejamento, revisão e retrospectiva de cada *sprint* uma vez por semana. Como o desenvolvimento era realizado apenas aos sábados e feriados, as reuniões diárias da equipe não existiram, limitando-se a conversas ocasionais e troca de mensagens através de aplicativo de mensagens instantâneas ou correio eletrônico.

##### 3.1.1 Pré-Game

Essa fase, dividida entre planejamento e arquitetura, começou com a visão do que deveria contemplar uma rede social para voluntariado. Após a visita de sites de voluntariado, apresentada na seção 2.1.6 Redes Sociais, a equipe discutiu e elaborou a seguinte lista de requisito:

- 1) Cadastro de Usuários (voluntários ou organizações);
- 2) Cadastro de Oportunidades de Voluntariado;
- 3) Visualização de uma *timeline*, contendo os *posts*, curtidas e comentários feitos pelos usuários;

- 4) Possibilidade de realizar, comentar e curtir *posts*;
- 5) Possibilidade de um voluntário se candidatar à uma oportunidade de voluntariado;
- 6) Permitir à instituição gerar um certificado de participação para o voluntário que trabalhou em uma oportunidade;
- 7) Permitir ao usuário conectar-se em outros usuários, criando assim amizades.

A partir dessa lista, realizou-se a reunião de planejamento, no início do semestre, com a presença do orientador.

#### 3.1.1.1 Planejamento

Na reunião de planejamento, a equipe responsável pelo desenvolvimento do projeto se apresentou ao orientador, conforme demonstrado no QUADRO 4.

QUADRO 4 - RECURSOS DO PROJETO

<b>Nome</b>	<b>Ingresso</b>
Ciro Alexandre Stahlschmidt Corrêa	30/07/2018
Juan Nicolas Varella	30/07/2018

FONTE: Os Autores (2018).

Após a apresentação da equipe inicial, foi discutida a visão do projeto, etapa da Engenharia de Sistema, e a partir dela foi elaborado o *backlog* do produto, registrando os requisitos do software, conforme representado no QUADRO 5.

QUADRO 5 - BACKLOG DO PRODUTO

**Backlog de Produto**

Voluntarius

Atualização: 14/08/2018

IED real diário: 26%

Sprint	id	Assunto	Item	Responsável	IED	Risco	Dias Estimados
1	1	Estrutura	Protótipo Login	Ciro	0,25	Baixo	0,34
1	2	Estrutura	Criar schema DB	Ciro	0,25	Baixo	0,34
1	3	Estrutura	Criar Projeto Java	Ciro	0,25	Baixo	0,34
2	4	Estrutura	Aplicar layout	Juan	0,25	Baixo	0,34
2	5	Estrutura	Criar Beans, DAOs, Controladora e Façade.	Ciro	0,25	Baixo	0,34
2	6	Cadastro	Criação de Usuário	Juan	0,25	Baixo	0,34
3	7	Cadastro	Cria Perfil do Voluntário	Juan	0,5	Médio	0,68
3	8	Processo	Ajustar Layout Timeline	Ciro	0,25	Baixo	0,34
4	9	Processo	Criar Timeline	Ciro	0,5	Médio	0,68
4	10	Cadastro	Adicionar Causas - Voluntário	Juan	0,25	Baixo	0,34
5	11	Processo	Criar Posts	Ciro	0,5	Médio	0,68
5	12	Cadastro	Adicionar Habilidades/Causas - Voluntário	Juan	0,25	Baixo	0,34
6	13	Cadastro	Perfil de Instituições	Juan	0,25	Baixo	0,34
6	14	Processo	Comentar Posts	Ciro	0,25	Baixo	0,34
7	15	Processo	Curtir Posts	Ciro	0,5	Médio	0,68
7	16	Cadastro	Editar Perfil Voluntários/Instituições	Juan	0,5	Médio	0,68
8	17	Processo	Incluir Posts com Imagens	Ciro	1	Alto	1,35
8	18	Consulta	Visualizar Perfil do Usuário	Juan	0,25	Baixo	0,34
9	19	Consulta	Pesquisar Usuários	Juan	0,25	Baixo	0,34
10	20	Consulta	Visualizar Instituições e Amigos do Voluntário	Juan	0,25	Baixo	0,34
10	21	Consulta	Visualizar Voluntários Seguidos pela Instituição	Juan	0,25	Baixo	0,34
11	22	Processo	Manter Oportunidade de Voluntariado	Ciro	0,5	Médio	0,68
12	23	Consulta	Pesquisar Oportunidades de Voluntariado	Ciro	0,5	Médio	0,68
13	24	Cadastro	Manter Amizade	Juan	1	Médio	1,35
14	25	Processo	Envio/Recebimento de Mensagens	Juan	0,5	Médio	0,68
14	26	Processo	Emitir Certificado de Participa	Ciro	1	Alto	1,35
Total:					10,75		15

FONTE: Adaptado de MARTINS; JOSÉ CARLOS CORDEIRO (2007).

Os requisitos foram distribuídos entre os responsáveis (integrantes da equipe no início do projeto) e classificados em níveis de risco baixo, médio ou alto.

Devido ao fato de os integrantes da equipe trabalharem e cursarem outras disciplinas, ficou estipulado entre eles que os trabalhos ocorreriam nos sábados e feriados, em um período de 8 horas. A partir disso, foi feito um cálculo de IED Real Diário utilizando a seguinte metodologia:

- 1) Quantidade de semanas no 2º semestre letivo de 2018 até a data limite de entrega do trabalho: 17 semanas, 3 semanas para documentação e 14 semanas para programação.

- 2) Das 14 semanas, 10 semanas são normais, restando 4 semanas com feriado em dia útil. Em termos percentuais, semanas normais representam 71% enquanto semanas com feriados representam 29%.
- 3) Dividindo-se um dia trabalhado (Sábado) entre os cinco dias de uma semana, obtemos o percentual de 20%. Se considerarmos dois dias trabalhados (um sábado e um feriado), obtemos o percentual de 40%.
- 4) Aplicando os percentuais obtidos no item 2 sobre os percentuais obtidos no item 3, respectivamente, obteremos:  $71\% \times 20\% = 14,20\%$  e  $29\% \times 40\% = 11,60\%$ , aproximadamente.
- 5) Somando-se os valores obtidos,  $(14,20\% + 11,60\%)$  chegamos ao percentual de 25,80%, arredondado na planilha para o valor de 26%.

Além da elaboração do *backlog* do produto, montou-se o cronograma do projeto (APÊNDICE B) e o quadro de riscos (QUADRO 6) com possíveis ações de controle para mitigá-los e conseguir realizar a entrega do software.

QUADRO 6 - RISCOS DO PROJETO

nr.	Descrição	Etapas Envolvida	Solução para o Risco	Probabilidade do Risco	Impacto
1	Disponibilidade de tempo dos recursos.	Todas as etapas	Analisar evolução do projeto e tirar férias do trabalho, caso haja necessidade. Buscar recurso adicional para integrar à equipe inicial.	Médio	Grave
2	Falta de conhecimento em alguma tecnologia.	Desenvolvimento	Identificar qual a dificuldade e buscar opções para contorná-las, seja buscando o conhecimento em algum professor, na internet, ou até mesmo substituindo a tecnologia envolvida por outra opção.	Médio	Moderado
3	Foco na elaboração do projeto.	Todas as etapas	Elaborar o projeto no mesmo espaço físico, com o objetivo dos integrantes se policiarem e cobrarem maior dedicação entre si.	Baixa	Grave
4	Computadores apresentarem defeitos.	Todas as etapas	Identificar se o defeito é de hardware ou de software. No primeiro caso, providenciar o conserto ou a substituição do computador. No segundo caso, buscar conhecimento para a correção do problema, considerando inclusive reinstalar o software.	Médio	Moderado

FONTE: Os Autores (2018).

Foram selecionadas as ferramentas de desenvolvimento e infraestrutura necessárias, todas elas apresentadas na seção 3.2.

Até aqui, ocorreram as Etapas de Engenharia de Sistema, Análise de Requisitos e Projeto Técnico, conforme explicado anteriormente.

### 3.1.2 Game

Nessa fase iniciou-se a fase de desenvolvimento do software, etapa de construção e validação do software, através da divisão das atividades em *Sprint* semanais, com as entregas sendo apresentadas semanalmente ao orientador. À medida que as apresentações eram realizadas, recebia-se um feedback sobre a evolução e planejava-se o *Sprint* seguinte.

#### 3.1.2.1 *Sprint* 1

No primeiro *Sprint*, foi solicitado pelo orientador que se apresentasse um protótipo da tela inicial do software, com o processo de autenticação funcionando. Para que isso fosse possível, criou-se o *Schema* TCC a as tabelas com seus devidos campos, conforme definido no Diagrama de Entidade Relacionamento (APÊNDICE F), utilizando a interface *MySQL Workbench*, para se comunicar com o banco de dados. Além disso, criou-se o projeto TCC no *Netbeans*, importando a biblioteca necessária (*Driver JDBC* do *MySQL*) para se comunicar com o banco de dados, bem com a biblioteca *JSTL* e a biblioteca *GSON* (utilização do *Ajax*).

Uma vez criada a estrutura, realizou-se uma tela de login funcional. Para isso, foi criado o arquivo da página principal do software (*portal.jsp*), contendo os campos de login e senha. Também foi criada a classe (*controladoraservlet.java*) que recebe os campos preenchidos através de parâmetros e solicita a verificação dos mesmos para outra classe criada (*Facade.java*). Essa, por sua vez, realiza a chamada para a classe *DAO* (*usuarioDAO.java*), também criada nesse *sprint*, que através de um método (*autenticarUsuario*) realiza a consulta no banco de dados e verifica se o usuário existe e a senha está correta, através de uma cláusula *SQL*. O leiaute foi realizado pela própria equipe, que percebeu a necessidade de buscar um template pronto com o objetivo de ganhar tempo e manter o foco no desenvolvimento da regra de negócio.

#### 3.1.2.2 *Sprint* 2

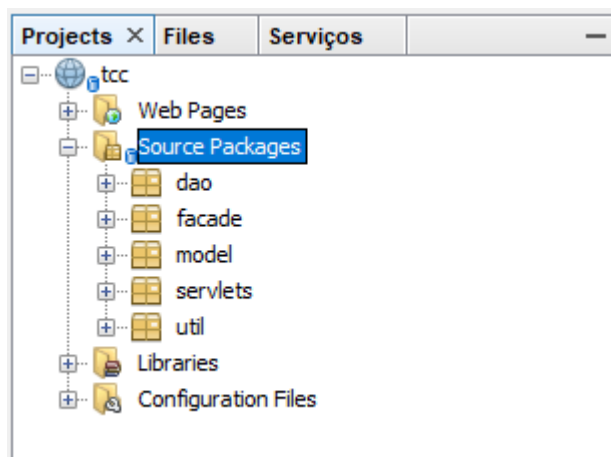
No segundo *Sprint*, foi apresentado o leiaute definitivo, obtido através do template adquirido. O template disponibilizou uma série de modelos de páginas *HTML*,



a serem escolhidas e adaptadas pela equipe à medida que as iterações ocorrerem. Também foram disponibilizados diversos arquivos CSS e *Javascript*, incluídos em suas respectivas pastas do projeto.

Foram criadas as estruturas de pacotes conforme ilustrado na FIGURA 23.

FIGURA 23 - PACOTES



FONTE: Os Autores (2018).

Dentro dos pacotes, foram realocadas as classes criadas para o processo de login. Foram criadas: as classes de negócios (*Beans*), conforme representadas no Diagrama de Classes (Apêndice D) e armazenadas no pacote “*model*”; as classes de comunicação com o banco de dados (DAOs), armazenadas no pacote “*dao*”; uma classe *Façade*, armazenada no pacote “*facade*”; uma classe Controladora *Servlet*, armazenada no pacote “*servlets*”; e uma classe chamada “Utilitarios”, responsável por criptografar a senha do usuário antes de ser gravada no banco de dados, armazenada no pacote “*util*”.

Na página principal, foi elaborada a funcionalidade de criar um novo usuário com os dados mínimos obrigatórios (nome, e-mail e senha). Caso o usuário seja uma instituição, há um *check-box* para clicar. Se marcado, ao processar a requisição do formulário, a controladora executa o método “*inserirInstituicao*” da *Façade*, que por sua vez executa o método de mesmo nome da classe “*InstituicaoDAO*”; caso fique desmarcado, é executado o método “*inserirVoluntario*”, que executa o método de mesmo nome da classe “*VoluntarioDAO*”.

### 3.1.2.3 Sprint 3

No terceiro *Sprint*, iniciou-se o desenvolvimento da página do perfil do voluntário (*perfil.jsp*), com foco na inclusão das informações complementares ao criar um usuário com esse perfil. Para isso, a controladora executa o método “*inserirVoluntario*” da *Façade*, que executa método de mesmo nome da classe “*VoluntarioDAO*”.

Também foi iniciado o desenvolvimento da página da *timeline* (*timeline.jsp*), com foco em readequar o leiaute HTML do *template*, uma vez que não havia um modelo exclusivo para a funcionalidade. Foi criada: uma *Façade* (*FacadePost*) e uma *Servlet* específica (*ControladoraPost*), para melhor organização do código, além de facilitar o desenvolvimento em conjunto pelos membros da equipe. Ao término do *Sprint*, chegou-se ao leiaute desejado, porém sem buscar nenhuma informação do banco de dados, exibindo o conteúdo oriundo do próprio *template*.

### 3.1.2.4 Sprint 4

No quarto *Sprint*, foi dada continuidade no desenvolvimento do perfil do voluntário, com foco na adição de causas que o usuário possui interesse de se dedicar a ações de voluntariado. Foram criados na classe DAO os métodos “*inserirVoluntarioCausa*” e “*excluiVoluntarioCausa*”.

No desenvolvimento da *timeline*, os posts foram incluídos manualmente no banco de dados e apresentados na página, utilizando JSTL e *Beans*, sem nenhuma interação por parte do usuário. Na classe “*PostDAO*”, foi criado o método “*listarPosts*”, responsável por listar as publicações do usuário e respectivos usuários com quem mantém amizade.

### 3.1.2.5 Sprint 5

No quinto *Sprint*, ainda houve trabalho na página do perfil do voluntário, incrementando a adição das causas em que o usuário defende ou possui interesse em ajudar. Para essa funcionalidade, foram criados os métodos “*inserirVoluntarioCausa*” e “*excluirVoluntarioCausa*” na classe DAO.

Quanto à *timeline*, foi implementada a funcionalidade de criar publicações (*posts*), eliminando dessa forma a necessidade de fazer o cadastro dos mesmos diretamente no banco de dados. Ao criar um post, é apresentado na página o nome do usuário e a data/hora da publicação. Para funcionar, foi criado o método “*inserirPost*” na classe *PostDAO*, que recebe a instância de um objeto *post* da “*FaçadePost*”, oriundo da controladora que recebeu os dados da requisição. Nessa funcionalidade, ficou a pendência de realizar uma publicação com imagem.

#### 3.1.2.6 *Sprint 6*

No sexto *Sprint*, iniciou-se o desenvolvimento da página do perfil de instituições (*perfillnst.jsp*), página diferente do perfil do voluntário pois naquela é definida apenas uma causa e não existem habilidades. A página utiliza o método “*consultarInstituicao*”, disponível na *Façade*, que executa método de mesmo nome da classe “*InstituicaoDAO*”, recebendo como parâmetro um objeto instituição.

Para a *timeline*, foi incrementada a opção de comentar uma publicação, identificando o nome do usuário e a data da publicação. Para essa funcionalidade, foi criado o método “*inserirComentario*” na classe “*PostDAO*”.

#### 3.1.2.7 *Sprint 7*

No sétimo *Sprint*, foi implementada a funcionalidade de curtir publicações, através do tradicional ícone de curtir, representado por uma mão com o polegar para cima, que quando clicado muda de cor e incrementa a quantidade de curtidas. Para implementá-la, foi utilizada a tecnologia AJAX através da criação da classe “*AjaxCurtirPost*”, para que a página não precise ser recarregada por inteiro, atualizando somente o ícone e o contador de curtidas. Além disso, foi criado o método “*curtirPost*” na classe *DAO* e na *Façade*.

Também foi implementada a opção de editar o cadastro de voluntário e instituições, através da criação dos métodos “*atualizarVoluntario*” e “*atualizarInstituicao*” nas respectivas classes *DAO*.

### 3.1.2.8 *Sprint 8*

No oitavo *Sprint*, foi desenvolvida a parte de criar publicações com imagens, através da criação do método “extractFileName”, que permite receber o arquivo enviado através do formulário de publicações.

Também foi implementada a funcionalidade de visualização do perfil do usuário, acessado quando usuários realizam buscas de voluntários ou instituições. Criados os métodos nas classes DAO e realizado ajuste na *Facade*.

### 3.1.2.9 *Sprint 9*

No nono *Sprint*, um membro da equipe parou de programar para dar maior atenção à elaboração deste documento.

. Na parte de desenvolvimento, foi dado início à funcionalidade de pesquisar instituições e voluntários.

### 3.1.2.10 *Sprint 10*

No décimo *Sprint*, foi dado início ao desenvolvimento das páginas que exibem quais voluntários são seguidos por uma instituição e quais instituições são seguidas por um voluntário (instituicoes.jsp e amigos.jsp). Também foi dado início à funcionalidade de enviar mensagem de um usuário para outro além de ajustar a definição de foto para o perfil do usuário.

### 3.1.2.11 *Sprint 11*

No décimo primeiro *Sprint*, iniciou-se o desenvolvimento de publicações contendo oportunidades de voluntariado. Ao publicar uma oportunidade, um voluntário poderá se candidatar a realização da mesma e, ao seu término, receber um certificado de participação.

#### 3.1.2.12 *Sprint* 12

No décimo segundo *Sprint*, foi iniciada a funcionalidade pesquisar oportunidades de voluntariado, através da criação da página (pesquisarOportunidade.jsp).

#### 3.1.2.13 *Sprint* 13

No décimo terceiro *Sprint*, foi implementada a funcionalidade de manutenção de amizade.

#### 3.1.2.14 *Sprint* 14

No décimo quarto *Sprint*, foi implementada a funcionalidade de emissão do certificado de participação do voluntariado e o processo de enviar mensagens entre os usuários.

#### 3.1.3 Pós-Game

Devido a metodologia ser adaptada ao desenvolvimento de um software em ambiente acadêmico, essa fase não teve uma forte demanda de trabalho da equipe. Uma vez que o software não será distribuído para clientes, muito menos feito um material para a área de marketing, o trabalho aqui limitou-se a elaborar um manual para a instalação do mesmo, pré-requisito para a entrega do trabalho e avaliação por parte da banca. O software e os documentos (manual e projeto) foram gravados em uma mídia entregue na secretaria do curso, dentro do prazo estipulado pela coordenação do curso.

### 3.2 MATERIAIS UTILIZADOS

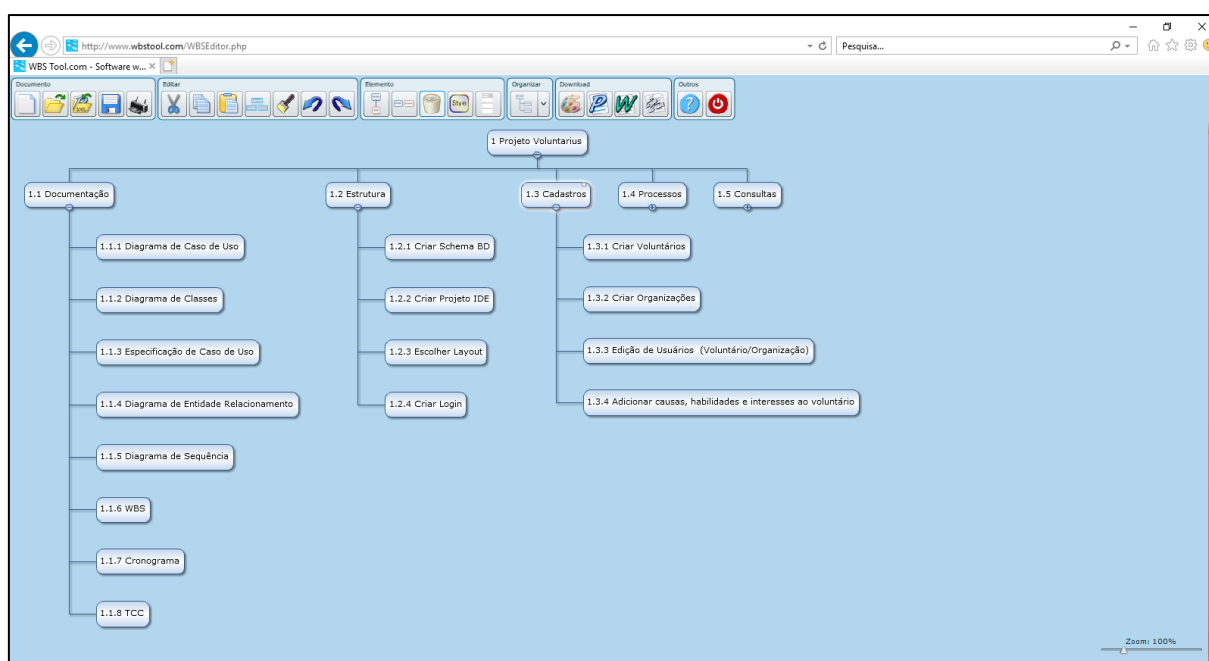
As ferramentas necessárias para a realização desse projeto foram: *Netbeans*, *MySQL*, *MySQL Workbench* e *Glassfish*, além dos computadores pessoais de cada integrante. Para o controle do projeto, foram utilizadas as ferramentas: *WBStool*, *Teamgantt*, planilhas e documentos *Google*, todas disponibilizadas gratuitamente na

internet. Foi obtido um *template* voltado para redes sociais. Para a elaboração dos diagramas da UML foi utilizado o software *Astah* e para a elaboração do diagrama entidade relacionamento foi utilizado o *brModelo*. Para a elaboração deste documento, foram utilizadas ferramentas desenvolvidas pela Microsoft®, o Microsoft Word e o *Paint*, além do *draw.io* disponibilizado pela *Google*.

### 3.2.1 WBSTool

É uma ferramenta gratuita e *online*, utilizada na etapa de planejamento para a elaboração da EAP (Estrutura Analítica do Projeto), tradução em português de WBS (*Work Breakdown Structure*). A FIGURA 24 apresenta a tela da ferramenta durante a elaboração da estrutura (APÊNDICE A).

FIGURA 24 - INTERFACE GRÁFICA WBSTOOL

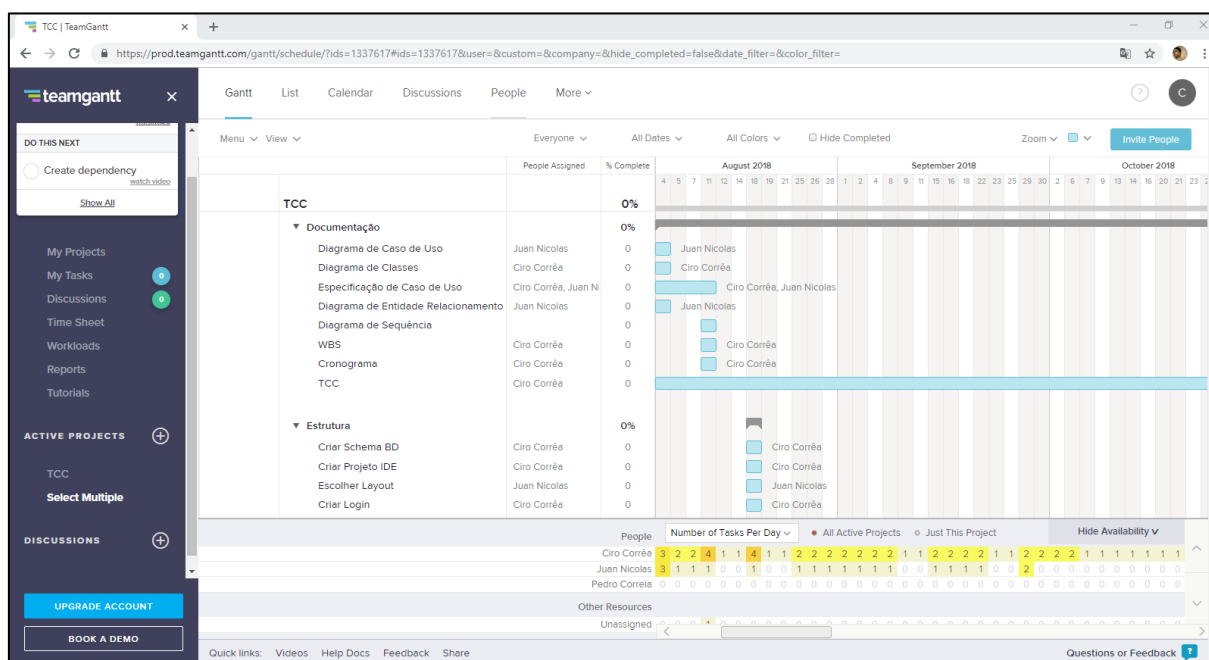


FONTE: Os Autores (2018).

### 3.2.2 Teamgantt

Também disponível de forma *online* e gratuita, o software foi utilizado durante a etapa de planejamento para a elaboração do cronograma (APÊNDICE B), conforme exibido pela FIGURA 25.

FIGURA 25 - INTERFACE GRÁFICA TEAMGANTT



FONTE: Os Autores (2018).

### 3.2.3 Template Redes Sociais

Com o intuito de obter um aspecto mais profissional na apresentação do software, a equipe optou por adquirir um *template* com aspecto de rede social. Foi adquirido através da internet, em uma pesquisa realizada pela equipe. O QUADRO 7 demonstra o investimento realizado.

QUADRO 7 - MATERIAL ADQUIRIDO

Material	Origem	Valor em R\$
Template Redes Sociais	<a href="https://bootdey.com">https://bootdey.com</a>	70,00

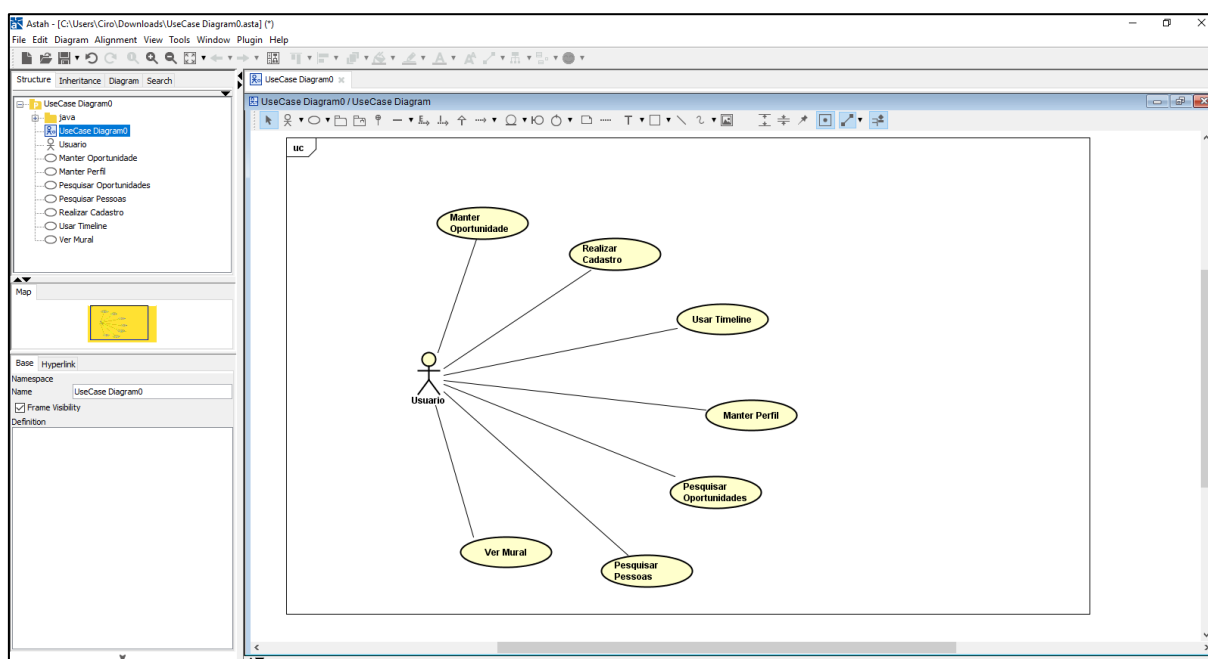
FONTE: Os Autores (2018).

### 3.2.4 Astah

Desenvolvido pela *Change Vision, Inc.*, o software é referência na construção dos diagramas utilizados na UML, mas não se limita a isso somente. Possui recurso para construção de fluxos e diagrama de entidade e relacionamento também.

É comercializado através de uma versão denominada *professional* e possui uma versão para fins acadêmicos, utilizada no projeto e denominada *Students*. Para obtê-la, é necessário que o estudante possua uma conta de e-mail em instituição de ensino reconhecida pela empresa. A FIGURA 26 ilustra a utilização do software, ainda na etapa de planejamento, durante a elaboração do Diagrama de Casos de Uso (APÊNDICE C), também utilizado para a elaboração do Diagrama de Classes (APÊNDICE E), Diagramas de Sequência (APÊNDICE F) e Diagrama de Estado (APÊNDICE G).

FIGURA 26 - INTERFACE GRÁFICA ASTAH



FONTE: Os Autores (2018).

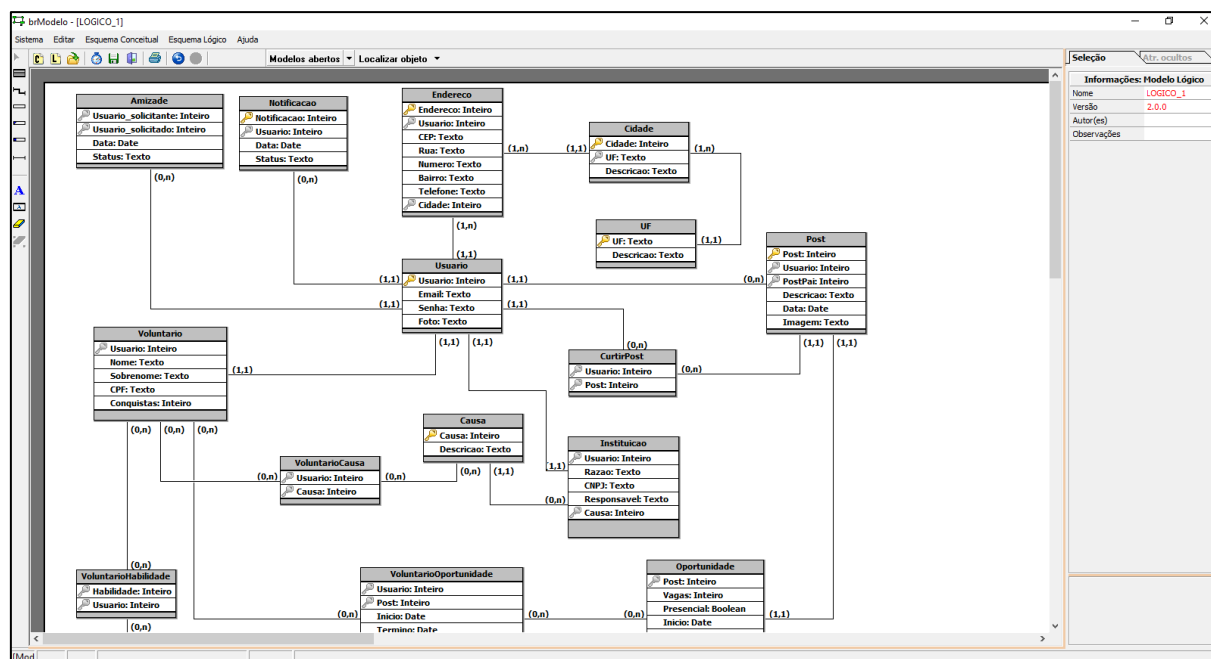
### 3.2.5 brModelo

Concebida em um trabalho de conclusão do curso de banco de dados pelas universidades UFSC (SC) e UNIVAG (MT), foi desenvolvida por Carlos A. Heuser com o objetivo de ensinar modelagem de banco de dados relacionais. É uma ferramenta de código aberto e totalmente gratuita, sua nova versão foi desenvolvida utilizando a linguagem Java, com entrega em abril de 2017, por Carlos Henrique Cândido, quem realizou melhorias. A FIGURA 27 ilustra a interface de utilização do software, também



na etapa de planejamento, durante a elaboração do Diagrama de Entidade Relacional do projeto (APÊNDICE H).

FIGURA 27 - INTERFACE GRÁFICA brMODELO



FONTE: SYS4.COM (2018).

### 3.2.6 Suítes Google e Microsoft

Criados pela *Google*® e disponibilizados de forma exclusivamente online e totalmente gratuita, Planilhas *Google* e Documentos *Google* são, respectivamente, uma opção de planilha de cálculos e editor de texto cujo pré-requisito para utilização é possuir uma conexão com a internet. A ferramenta Documentos *Google* foi utilizada para a elaboração das Especificações de Caso de Uso (APÊNDICE D), enquanto a ferramenta Planilhas *Google* foi utilizada para elaborar um rascunho inicial do cronograma do projeto e para tabular o resultado pesquisa de redes sociais de voluntariado existentes na internet. Os arquivos produzidos foram gravados no *Googledrive* (nuvem da *Google*) cuja vantagem é permitir o compartilhamento com outros usuários e possibilitar a edição por mais de um usuário ao mesmo tempo, mas também podem ser copiados para o computador, necessitando de um software que permita a visualização conforme o formato selecionado. A *Google* disponibiliza

aplicativos elaborados por terceiros, como por exemplo, o *draw.io*, desenvolvido pela *JGraph Ltda.*, utilizado para desenhar algumas figuras deste trabalho.

O Office365, comercializado pela *Microsoft®*, é uma das suítes precursoras e mais utilizada no mundo, principalmente no ambiente corporativo, onde se destaca o produto *Excel*, produto voltado para elaboração de planilhas de cálculos. A suíte é disponibilizada através de instalação no computador que irá utilizá-la, mas também possui uma forma *online* e gratuita, bastando ter uma conta na *Microsoft®*, porém com menos recursos do que a versão paga. Foi utilizado o editor de texto *Word* para a elaboração deste documento por funcionar *offline* e possuir recursos que facilitam a normatização do trabalho segundo as normas da ABNT, armazenando o arquivo na nuvem fornecida pela empresa, conhecida por *Onedrive*. Para ajustar as figuras aqui apresentadas, utilizando o *Paint* por ser uma ferramenta já disponível no ambiente *Windows*.

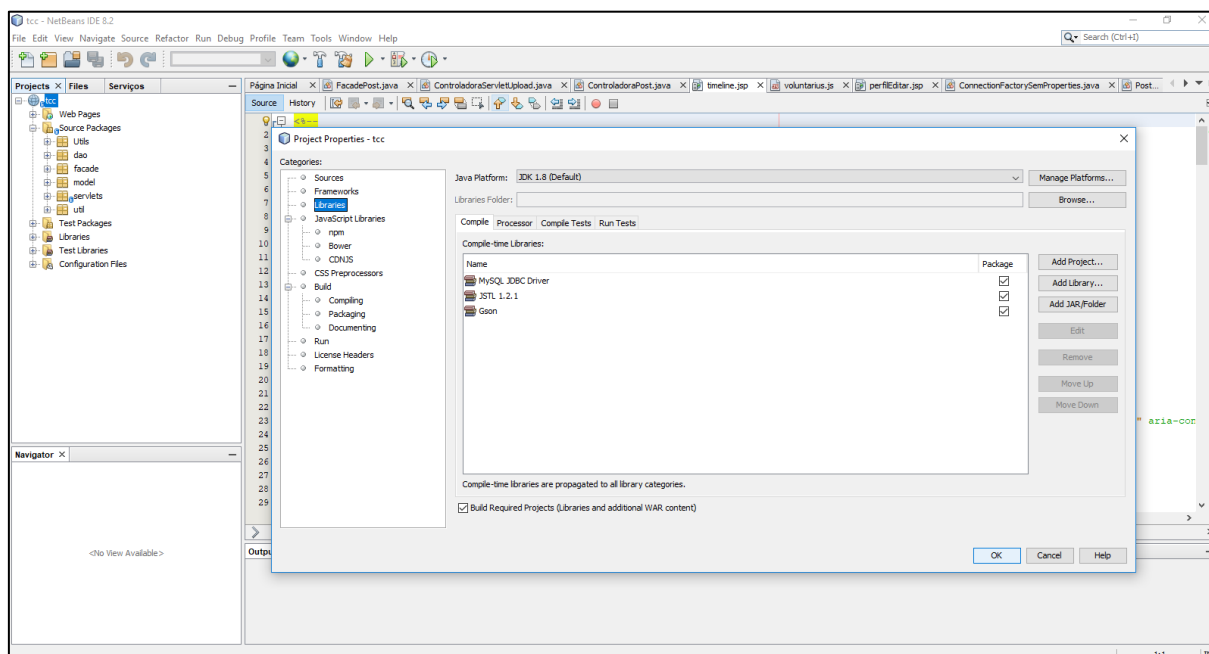
### 3.2.7 Netbeans

Disponibilizada de forma *open-source* pela Sun Microsystems® em junho de 2019, é uma IDE (*Integrated Development Environment*) e a principal ferramenta utilizada para o desenvolvimento do software. Bastante utilizada para escrever na linguagem Java, possui recursos para desenvolvimento em outras linguagens de programação, como por exemplo, PHP e C++, entre outras. Com mais de 18 milhões de *downloads*, possui mais de 800.000 desenvolvedores participando de sua comunidade, bastando se registrar de forma gratuita para participar.

É disponibilizada para download em seis versões, sendo Java SE a versão mais simples, comportando somente a linguagem Java para desenvolvimento de software para desktops. A versão completa, denominada JAVA EE, inclui não só suporte à linguagem Java, como outras linguagens também, além dos servidores embutidos, *Apache* e *Glassfish*, utilizados no desenvolvimento de aplicações *web* e corporativas, respectivamente.

No desenvolvimento da rede social *Voluntarius*, foi utilizada a versão JAVA EE com o servidor *Glassfish*, para suportar as tecnologias EJB, JSP, JSTL e SERVLETS. A FIGURA 28 exhibe o projeto criado na IDE, durante a fase *Game*, com a importação das bibliotecas utilizadas.

FIGURA 28 - INTERFACE GRÁFICA NETBEANS - BIBLIOTECAS



FONTE: Os Autores (2018).

### 3.2.8 MySQL

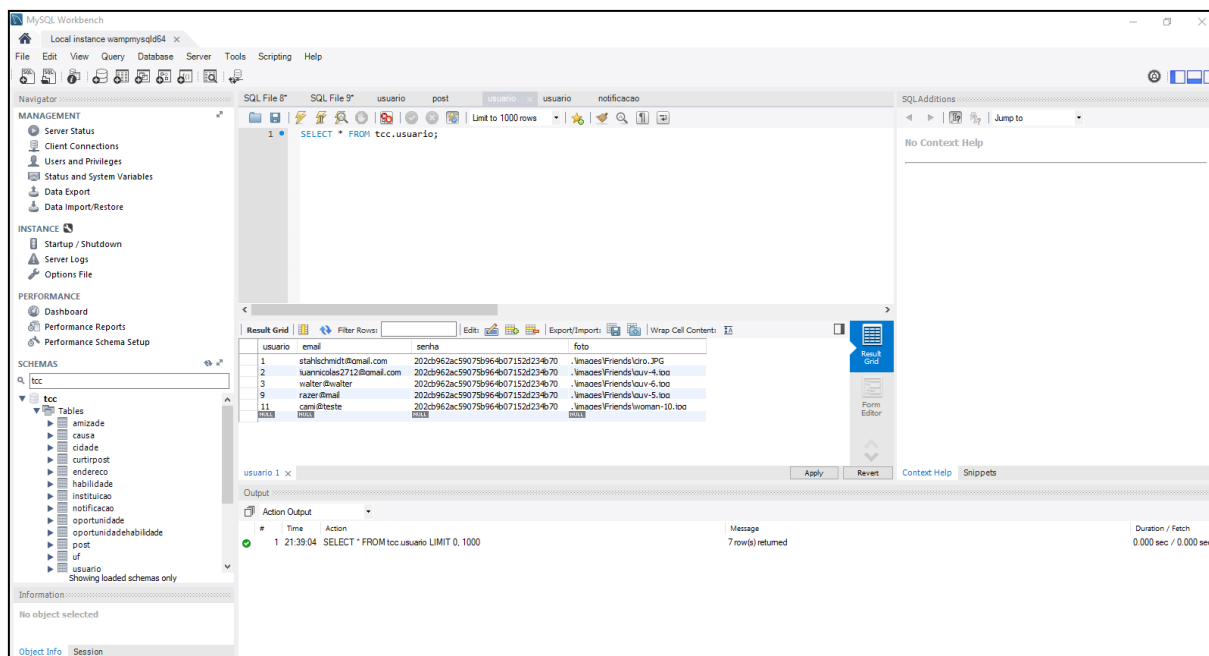
O banco de dados escolhido para a realização de nosso trabalho foi o MySQL™, atualmente mantido pela Oracle®, que pode ser distribuído de três formas:

- 1) *MySQL Community Server*: versão gratuita, fornecida sobre os termos da GPL (*General Public Licence*).
- 2) *MySQL Enterprise Edition*: versão comercial, com vários recursos adicionais e ferramentas de gerenciamento.
- 3) *Oracle MySQL Cloud Service*: Utilizando O MySQL Enterprise, é um serviço de hospedagem da Oracle, mantendo o banco de dados na nuvem.

Para esse trabalho, optou-se por utilizar a versão *MySQL Community*, por ser uma versão livre para a comunidade de desenvolvedores. O banco de dados foi instalado na máquina de cada integrante e para realizar a manutenção do mesmo, foi utilizado o software MySQL WorkBench, uma interface gráfica gratuita, utilizada para

criar um *schema* denominado TCC bem como a criação e manutenção das tabelas necessárias para o trabalho, conforme verificado na FIGURA 29.

FIGURA 29 - INTERFACE GRÁFICA MYSQL WORKBENCH

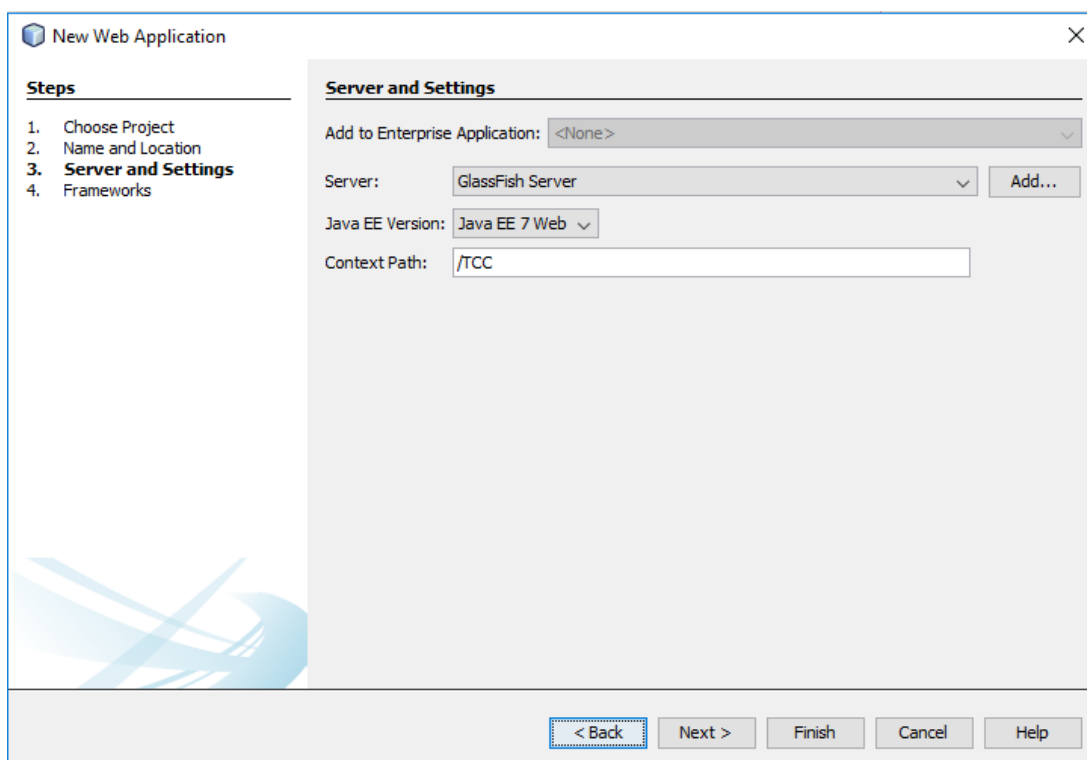


FONTE: Os Autores (2018).

### 3.2.9 Glassfish

Criado em 2005 pela Sun Microsystems®, com o objetivo de ser um servidor de aplicação totalmente certificado para a versão Java EE, herdou boa parte do código do Tomcat, doado pela Sun e JServ Group para a Apache (GONCALVES, 2010). Ao criar um projeto JAVA EE, é necessário definir qual será o servidor de aplicação Java responsável por traduzir as informações para a linguagem. A FIGURA 30 demonstra o momento em que o *Glassfish* é selecionado como servidor de aplicação, na criação do projeto.

FIGURA 30 - ESCOLHA DO GLASSFISH



FONTE: Os Autores (2018).

### 3.2.10 Hardware

Para o desenvolvimento do software, foram utilizados três computadores com especificação conforme demonstrada no QUADRO 8.

QUADRO 8- HARDWARE

<b>Fabricante:</b>	Avell	Sony
<b>Modelo:</b>	Titanium B155 Fire	Vaio SVF14A
<b>Processador:</b>	2,40 GHz Intel Core i3	1,80 GHz Intel Core i5
<b>Memória:</b>	4 GB	10GB

FONTE: Os Autores (2018).

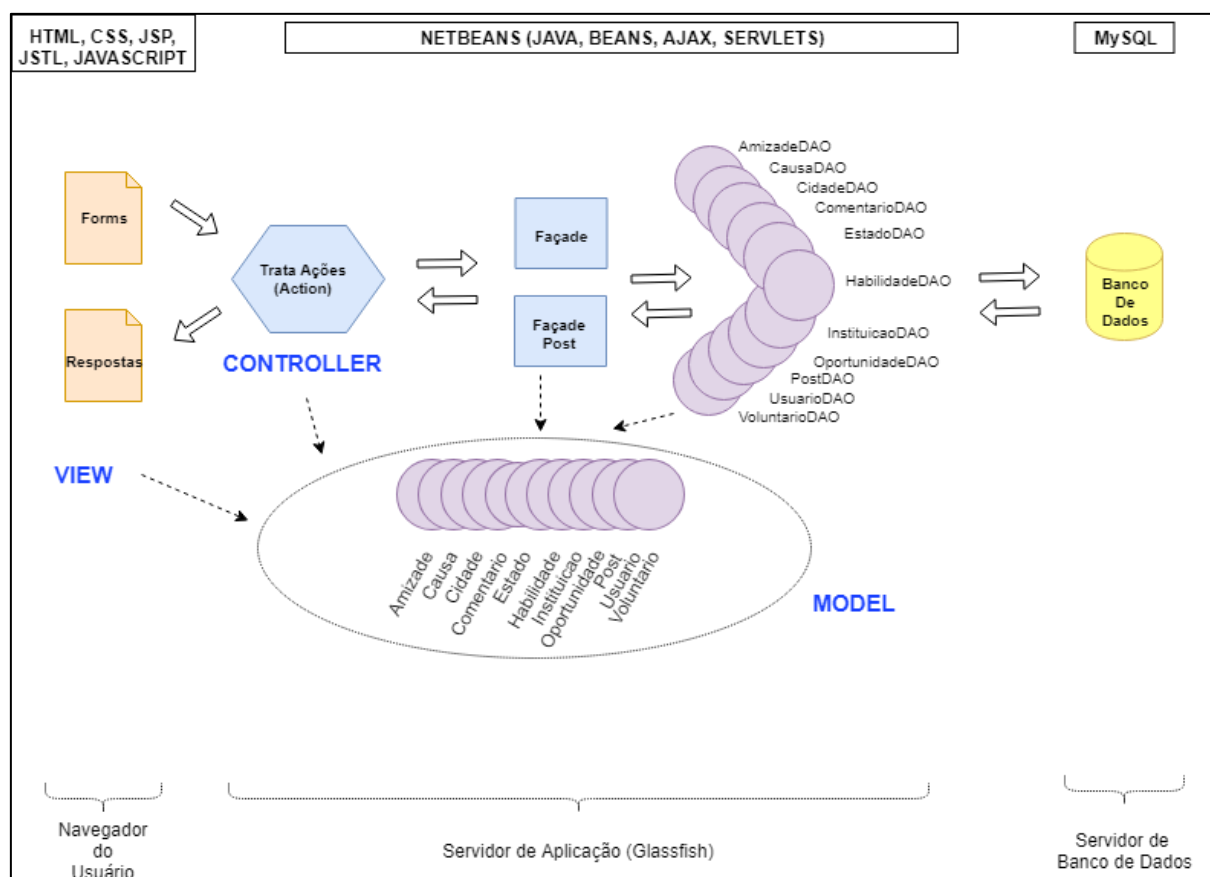
## 4 APRESENTAÇÃO DO SISTEMA

Apresentamos nesse capítulo, a arquitetura do software desenvolvido e uma breve explicação de como ele funciona, acompanhada pelas respectivas telas.

### 4.1 ARQUITETURA DO SOFTWARE

Após planejar o projeto, realizou-se uma revisão nas documentações e com base nas ferramentas a serem utilizadas para o desenvolvimento, foi possível definir a arquitetura do software ilustrada pela FIGURA 31 correlacionada com as ferramentas e tecnologias utilizadas.

FIGURA 31 - ARQUITETURA



FONTE: Os Autores (2018).

Percebe-se através da FIGURA as diversas camadas existentes na construção do software, seguindo o padrão MVC (*Model-View-Controller*) e utilizando as classes de Façade e DAO, todos representados no APÊNDICE E.

Na primeira coluna, as linguagens HTML, CSS, e Javascript e o componente JSTL estão inseridos dentro dos arquivos JSP, constituindo a parte visual (*view*), percebida pelo usuário ao acessar o sistema.

Qualquer ação realizada pelo usuário no software, inclusive o simples fato de acessar a página principal, é interpretada pelo servidor de aplicação (*controller*), que realiza toda a conversão em linguagem *Java*, acionando as *Servlets*, instanciando os *Beans* e, através do *Ajax*, atualizando somente parte da página, quando necessário.

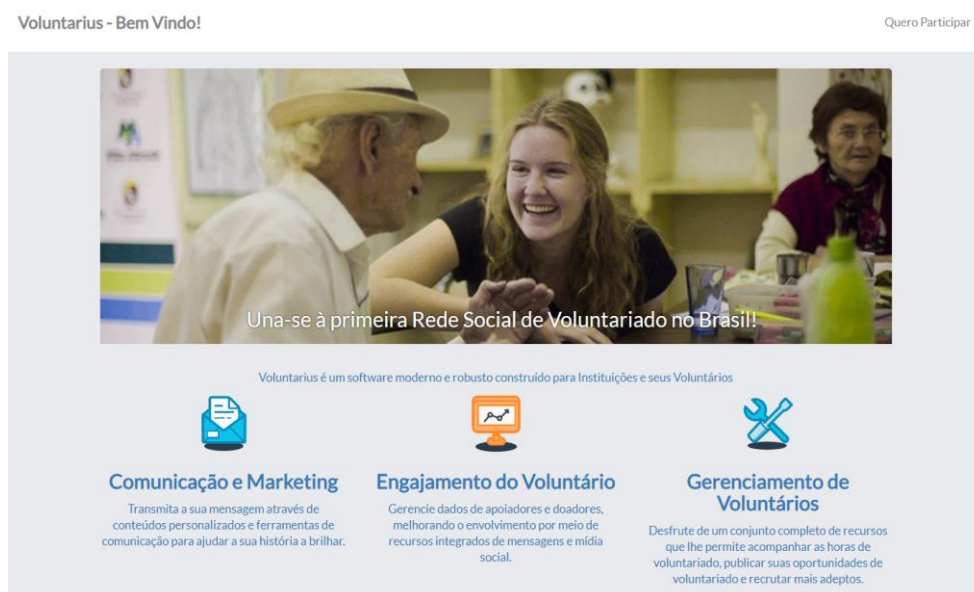
Toda solicitação realizada pelo servidor de aplicação ao banco de dados, é respondida através das classes *façade*, que por sua vez se comunicam com as classes *DAO* para a realização de comunicação com o banco de dados e assim persistir os dados no banco.

As classes de negócios (*model*) são utilizadas ao longo de todo esse processo descrito acima, toda vez que se faz necessário instanciar um objeto conforme a operação solicitada pelo usuário.

## 4.2 SOFTWARE

Na página inicial, é feita uma apresentação da rede social que consiste em informar ao usuário o propósito e o funcionamento da plataforma conforme apresentado na FIGURA 32.

FIGURA 32 - PÁGINA INICIAL



FONTE: Os Autores (2018).

Após clicar em “Quero Participar”, o usuário é redirecionado para a página de acesso, onde pode informar seu usuário e senha ou, caso não tenha se cadastrado ainda, realizar um pré-cadastro para obter os dados de acesso, conforme pode ser observado na FIGURA 33. Ao se cadastrar, o usuário instituição se diferencia do usuário voluntário marcando a opção “Quero cadastrar uma instituição”. Após preencher os campos e clicar em “Cadastrar”, o sistema registra os dados do usuário e atualiza a tela para que o usuário realize o acesso.

FIGURA 33 - PÁGINA DE ACESSO / CADASTRO

A interface da página de acesso e cadastro é apresentada sobre um fundo decorativo composto por uma grade densa de pequenas imagens de perfil de diversos usuários. O formulário centralizado é dividido em duas seções principais:

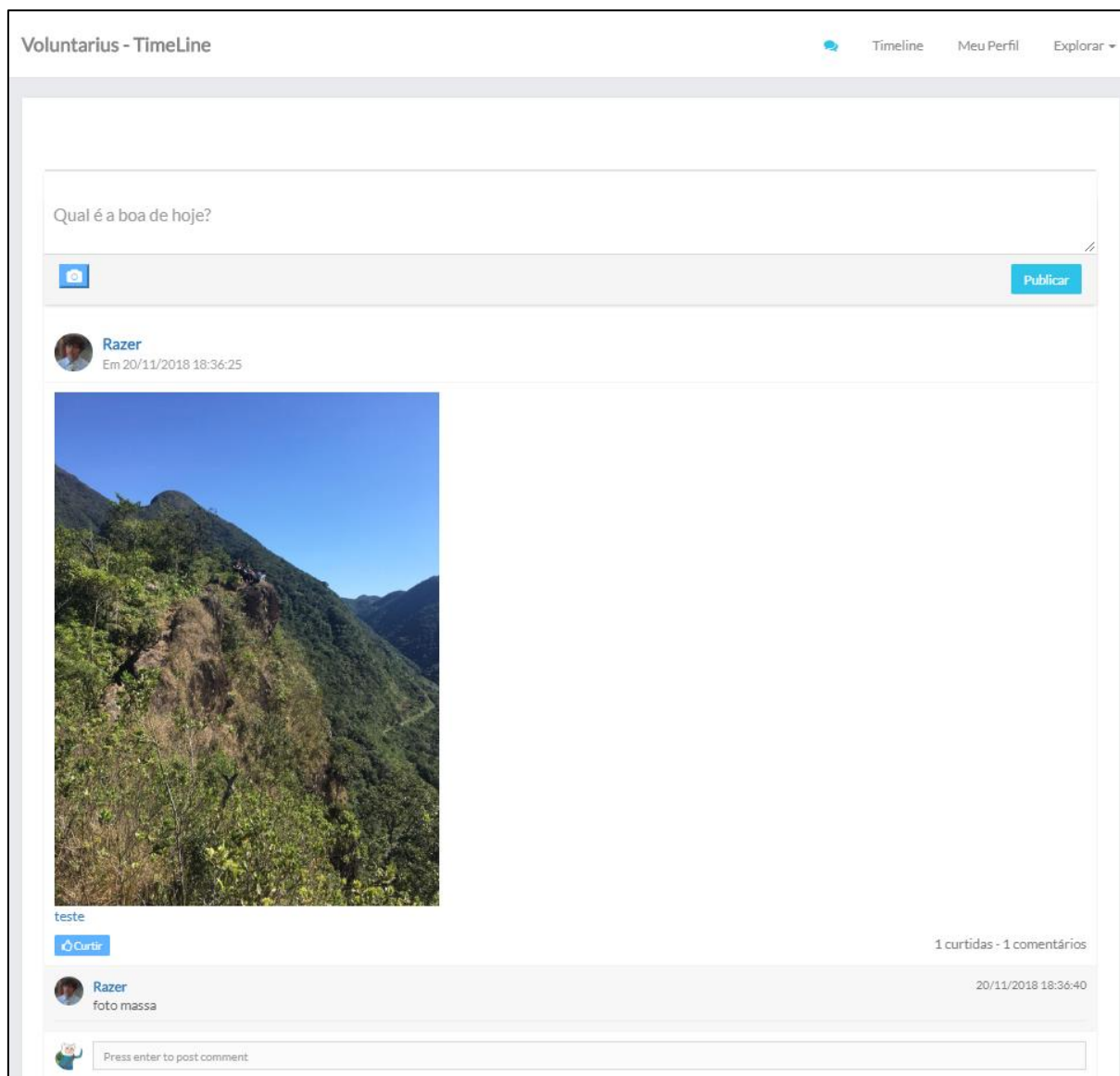
- Login:** Contém o subtítulo "Acesse sua conta" e dois campos de entrada para "Endereço de Email" e "Senha". Abaixo dos campos, há um botão azul com o texto "Acessar".
- Cadastre-se:** Contém o subtítulo "Crie uma nova conta" e campos para "Nome Completo", "Endereço de Email" e "Senha". Abaixo dos campos, há uma opção de seleção com o texto "Quero cadastrar uma instituição" e um botão azul com o texto "Cadastrar".

FONTE: Os Autores (2018).

Após a realização do login, o usuário é direcionado para a página de *timeline* da plataforma, onde é possível incluir *posts*, realizar comentários e curtir outros, bem como visualizar oportunidade de voluntariado. No primeiro acesso, a tela é apresentada em branco e à medida que o usuário interage com os demais, é apresentada conforme FIGURA 34.



FIGURA 34 - TIMELINE



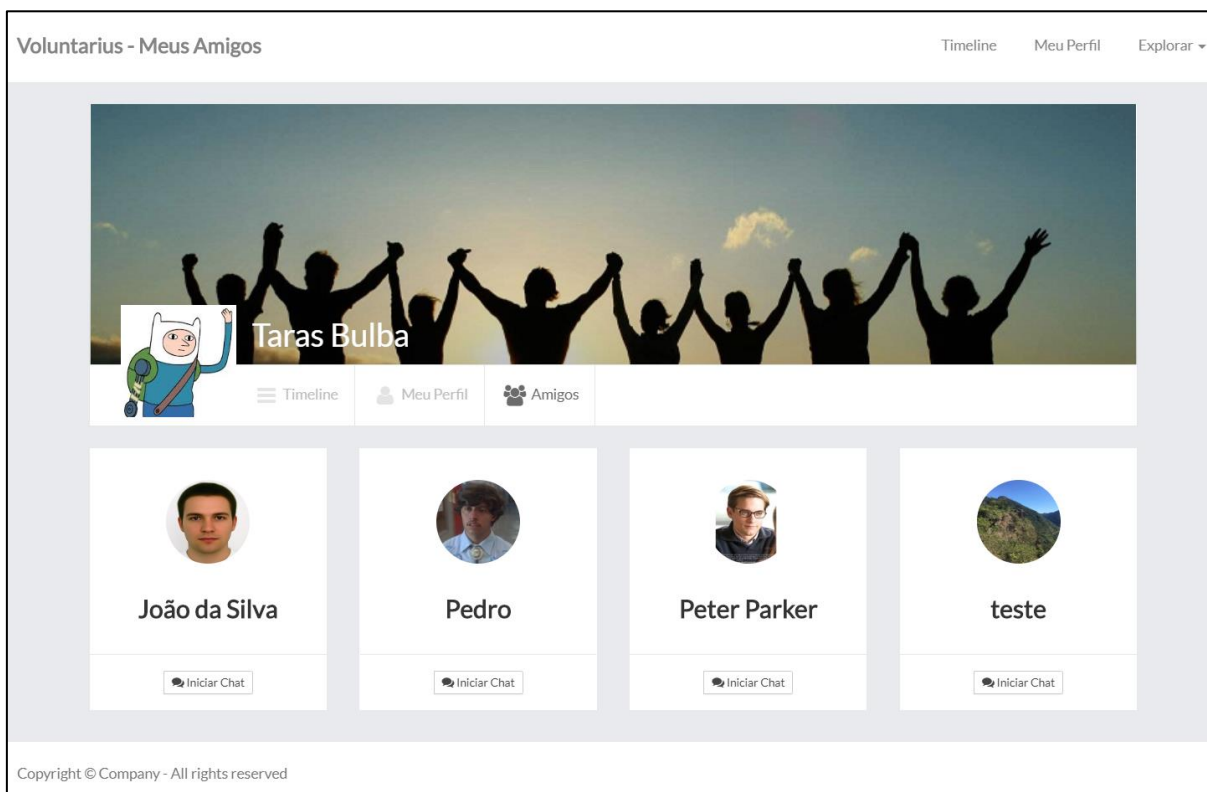
FONTE: Os Autores (2018).

Através do menu de acesso, o usuário pode visualizar seu perfil cadastro bem como editar as informações. Caso seja um usuário do tipo voluntário, a tela é apresentada conforme FIGURA 35. Nela é possível informar até três causas e até três habilidades para o perfil. Além disso, pode informar os dados pessoais (endereço e CPF), bem como sua experiência como voluntário.



De forma similar, ao clicar no link Amigos, é apresentada a pagina conforme FIGURA 37.

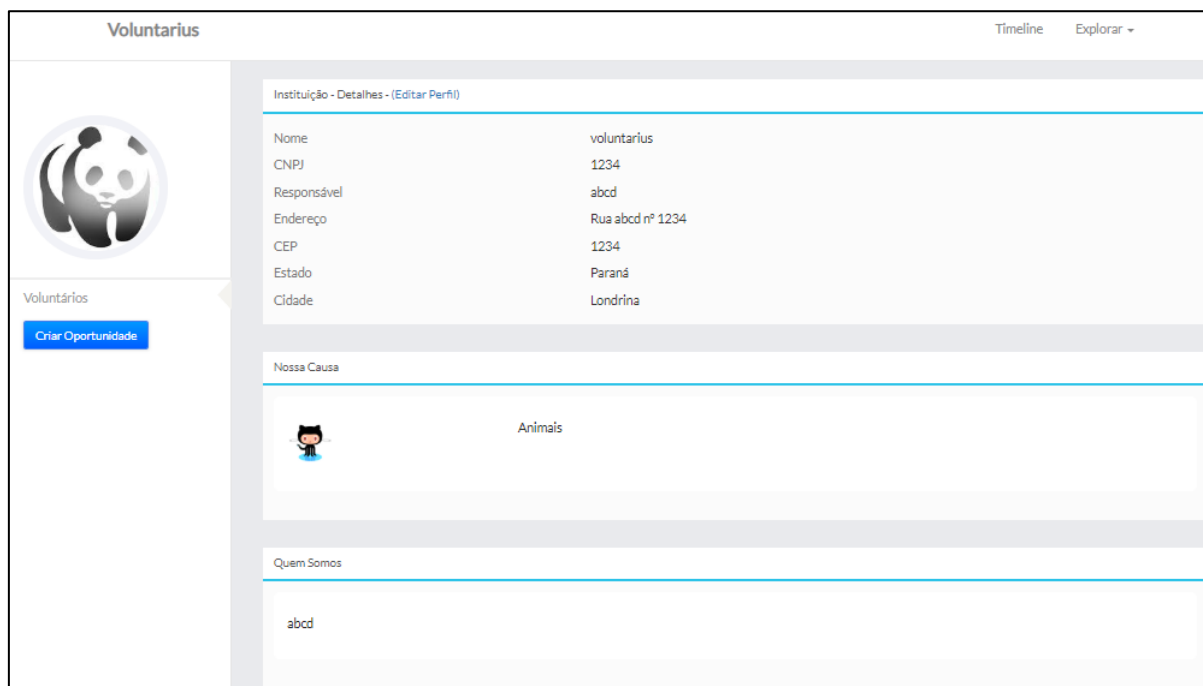
FIGURA 37 - MEUS AMIGOS



FONTE: Os Autores (2018).

Caso seja um usuário do tipo Instituição, a página de perfil é apresentada conforme a FIGURA 38. Nela é possível informar apenas uma causa e uma apresentação, além do endereço.

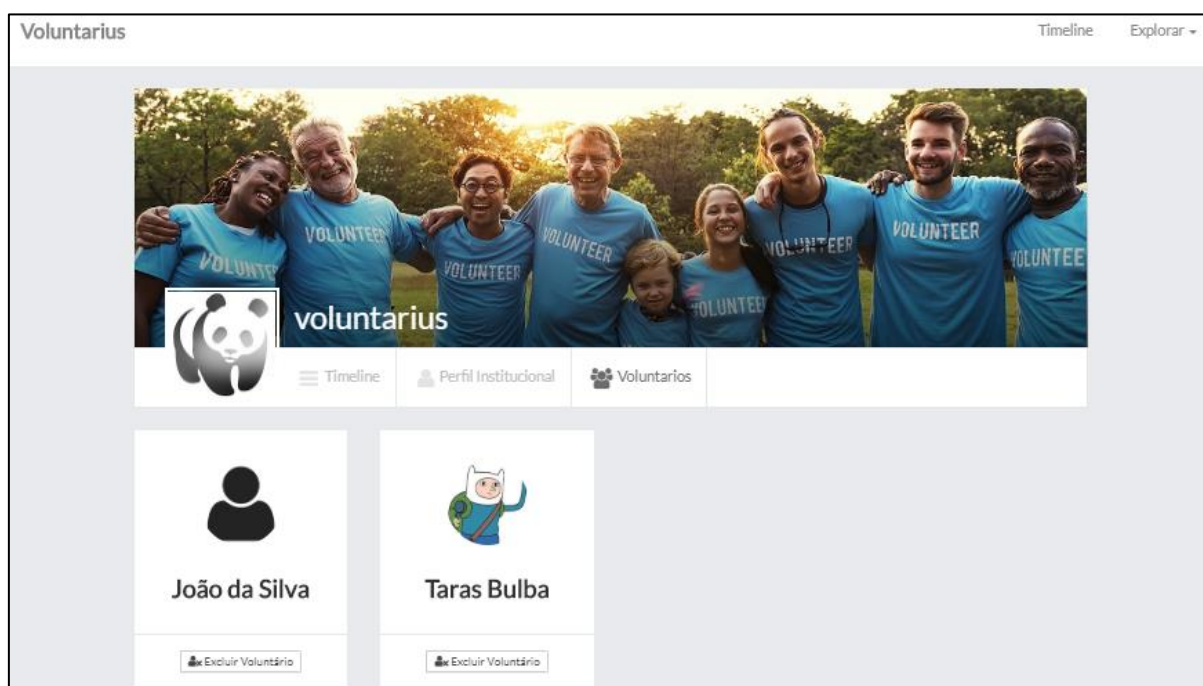
FIGURA 38 - PERFIL INSTITUIÇÕES



FONTE: Os Autores (2018).

Ao clicar no link Voluntários, é apresentada a página dos voluntários que seguem a instituição, conforme FIGURA 39.

FIGURA 39 - MEUS VOLUNTÁRIOS



FONTE: Os Autores (2018).

O outro *link* disponível na página da instituição é para criar uma oportunidade de voluntariado, que direciona para a página representada na FIGURA 40. É necessário que a instituição informe uma descrição para o título da oportunidade, se a mesma necessita ser realizada presencialmente ou não, a data de início e de finalização, para que os voluntários possam se programar, bem como a quantidade de vagas disponíveis e a carga horária a ser cumprida por cada voluntário. Além disso, é possível informar até três habilidades requeridas para o serviço voluntário da oportunidade. Pode-se informar uma imagem e, ao término de sua gravação, a oportunidade é gravada com status “Editada”, o que permite a instituição realizar alguma alteração. A oportunidade só estará disponível para visualização por outros usuários a partir do momento que for publicada. Uma vez publicada, a oportunidade será apresentada na *timeline*, para que os usuários possam se voluntariar.

FIGURA 40 - CRIAR OPORTUNIDADE

Voluntarius

Timeline Explorar

Criar Oportunidade

Descrição

Título da Oportunidade

Presencial

☒ Sim

☐ Não

Alterar

Data de Início

Data de Finalização

Vagas

Indique a quantidade de vagas

Carga Horária

Indique a quantidade de horas totais desta oportunidade

Status

Habilidade 1

Habilidade 2

Habilidade 3

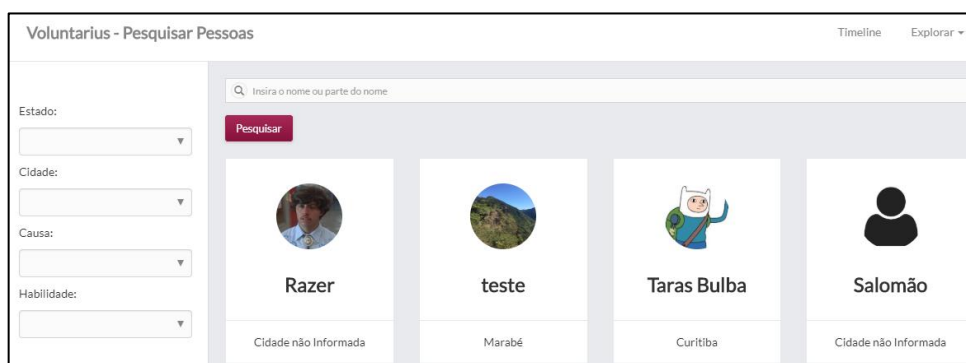
Gravar

Copyright © Company - All rights reserved

FONTE: Os Autores (2018).

No menu do portal, é possível realizar as pesquisas de voluntários, instituições e oportunidades. Na tela de pesquisas de voluntários, representada pela FIGURA 41, é possível buscar voluntários por estado, cidade, causas e habilidades. Caso o usuário deixe o filtro em branco, todos os registros serão exibidos.

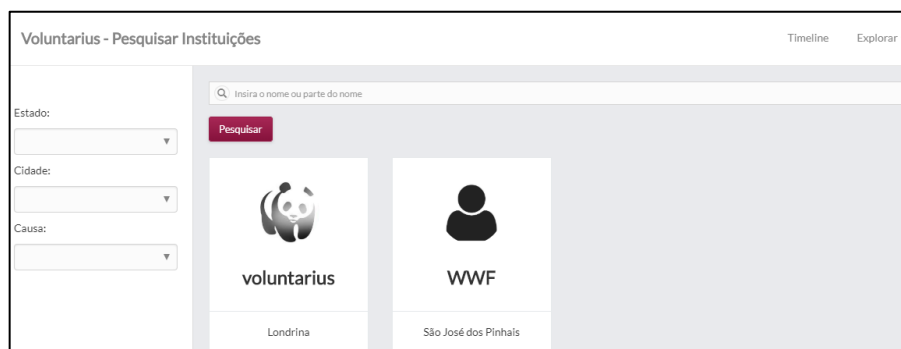
FIGURA 41 - PESQUISAR VOLUNTÁRIOS



FONTE: Os Autores (2018).

A tela de pesquisa de instituições, demonstrada na FIGURA 42, permite aos usuários realizarem uma busca por estado, cidade e causa. Caso o usuário deixa os filtros em branco, todos os registros serão exibidos.

FIGURA 42 - PESQUISAR INSTITUIÇÕES



FONTE: Os Autores (2018).

A tela de pesquisa de oportunidades permite aos usuários realizarem a busca por oportunidades de voluntariado, filtrando por estado, cidade, causa, habilidades, necessidade de presença, além de permitir filtrar em quais oportunidades o usuário está inscrito. Uma vez listadas as oportunidades disponíveis, é possível clicar para visualizar maiores informações sobre a oportunidade escolhida. A FIGURA 43 exibe uma pesquisa de todos os registrados, com o filtro em branco.

FIGURA 43 - PESQUISAR OPORTUNIDADES

Voluntarius - Pesquisar Oportunidades

Timeline Meu Perfil Explorar ▾

Estado:

Cidade:

Causa:

Habilidade:

Presencial:

☐ Sim




☐ Não

☒ Ambos

☐ Minhas Oportunidades

Insira o título da oportunidade de voluntariado ou o nome da org anfitriã.

Pesquisar

 <p>voluntarius <b>teste0411</b> 18/11/2018</p>	 <p>voluntarius <b>TesteRazer 1</b> 08/11/2018</p>	 <p>voluntarius <b>teste0411.2</b> 04/11/2018</p>
--	---	--

Copyright © Company - All rights reserved

FONTE: Os Autores (2018).


Ao clicar em uma oportunidade encontrada, o usuário é direcionado para uma página contendo as informações cadastradas pela instituição responsável pela mesma. Caso a oportunidade seja visualizada por um usuário qualquer, é apresentado um botão “Quero Participar”, que ao ser clicado informa ao usuário que sua solicitação será analisada. Se a própria instituição visualizar a oportunidade, são apresentados botões para editar a oportunidade e listar os voluntários que dela estão participando, conforme representado na FIGURA 44.

FIGURA 44 - VISUALIZAR OPORTUNIDADE

Voluntarius - Oportunidade - Visualizar Timeline Explorar ▾




Criar Oportunidade

Descrição	teste
Presencial	Sim



Data de Início	16/11/2018
Data de Finalização	16/11/2018
Vagas	3
Carga Horária	4
Status	Publicada

Habilidade(s) Requerida(s)

	Artista
	Música
	Matemática

[Editar Oportunidade](#) [Listar Voluntarios](#)

FONTE: Os Autores (2018).





Caso a instituição clique no botão “Editar Oportunidade”, o usuário é redirecionado para a página de edição da mesma, igual à tela de criação de oportunidade, porém com os dados cadastrados previamente sendo apresentados.

Se o botão “Listar Voluntários” for clicado, o usuário é redirecionado para a página demonstrada na FIGURA 45, exibindo os voluntários que dela solicitaram participar, bem como voluntários que já estão participando na oportunidade em questão. É possível a instituição clicar em um botão para editar as informações.



FIGURA 45 - LISTAR VOLUNTÁRIOS

Voluntarius Timeline Explorar ▼


Voluntário	Início	Término	Status	Horas	Editar
 Taras Bulba - Curitiba/PR	21/11/2018	21/11/2018	Aceito	6	
 teste - Marabá/PA			Rejeitado	0	

FONTE: Os Autores (2018).

Uma vez clicado no botão Editar de um determinado voluntário, o usuário é direcionado para a página exibida na FIGURA 46, onde a instituição pode aceitar, rejeitar ou finalizar a participação do voluntário, sendo necessário, nesse último caso, preencher os campos de data de término e a quantidade de horas realizadas pelo mesmo.

FIGURA 46 - EDITAR VOLUNTÁRIO

Voluntarius Timeline Explorar ▼

Voluntário	Início	Término	Status	Horas
 <b>Taras Bulba</b> - Curitiba/PR	<input type="text" value="21/11/2018"/>	<input type="text" value="21/11/2018"/>	<input type="button" value="Aceitar"/>	<input type="text" value="6"/>

FONTE: Os Autores (2018).

Ao finalizar a oportunidade de um voluntário, o portal disponibiliza ao mesmo a geração de um certificado, contendo seu nome, o nome da instituição, o período e a carga horária total realizada, servindo como forma de comprovante para o voluntário do serviço prestado para a instituição. A FIGURA 47 exibe um certificado gerado pelo portal.

FIGURA 47 - CERTIFICADO



FONTE: Os Autores (2018).

## 5 CONSIDERAÇÕES FINAIS

O objetivo deste trabalho foi desenvolver uma rede social de voluntariado. Com os conhecimentos adquiridos ao longo do curso e após uma pesquisa na internet, percebeu-se que os sites de voluntariado não permitem que voluntários e instituições interajam entre si, limitando-se a exibir oportunidades de voluntariado.

Através da rede social desenvolvida, foi disponibilizado ao voluntário informar suas habilidades e quais causas o estimulam a se voluntariar, além de poder escrever um texto livre contendo seus interesses pessoais. Por outro lado, foi disponibilizado às instituições, definir qual causa abraçam e publicar oportunidades de voluntariado para que candidatos se inscrevam e participem, contribuindo para a demanda em questão.

Tanto para o voluntário quanto para a instituição, foram disponibilizados filtros de busca que auxiliem ambos a se encontrarem e realizarem um laço de amizade. Ao realizar a publicação de uma oportunidade de voluntariado, a rede social permite ao voluntário analisar a demanda da instituição e se candidatar, caso a atividade esteja dentro de suas possibilidades. Informações como local da atividade, duração, necessidade da presença e carga horária total ajudam na tomada de decisão. Ainda que falte alguma informação adicional, é possível enviar uma mensagem pela própria plataforma ou entrar em contato através dos meios de comunicação disponibilizados pela instituição.

Uma vez aprovada pela instituição a adesão do voluntário à oportunidade de voluntariado, a plataforma gera um certificado quando chegar ao término da atividade. Esse foi outro benefício idealizado, com o intuito de facilitar para o voluntário que busca atividades desse nicho com o intuito de comprovar sua dedicação para fins próprios, como por exemplo, cumprir horas formativas em cursos de graduação.

Outro benefício da plataforma é permitir aos usuários, sejam instituições ou voluntários, curtir e comentar publicações realizadas por outros usuários. Por ser uma plataforma voltada para atividades de voluntariado, pode-se dizer que outro benefício para as instituições seria trazer maior visibilidade nas necessidades, que podem ser necessidades de outras instituições, permitindo assim uma troca de experiência.

Na área de tecnologia, o desenvolvimento da plataforma permitiu à equipe consolidar conhecimentos adquiridos e identificar outros que precisam ser melhorados, devido às dificuldades naturais que surgiram ao longo dessa caminhada

e que estimulam o constante processo de aprendizagem, através da busca do conhecimento.

Ao longo do trabalho, também foi percebido um enriquecimento de conhecimento em outras áreas, que não a tecnologia propriamente dita. Em um projeto, onde participam pessoas e ideias diferentes, é natural o surgimento de conflitos e aprender a administrá-los é de grande valia para a vida profissional, pois nem sempre estaremos lidando com pessoas que pensam da mesma maneira, o que nos enriquece como seres humanos, ao perceber uma outra realidade que não a nossa.

Outra área de aprendizado, foi aprender a lidar com o tempo. Estimar o tempo necessário para a elaboração de um software, mesmo com todas as ferramentas de gestão existentes, sempre estará sujeito a algum risco, que poderá prejudicar sua conclusão e às vezes até mesmo paralisar um projeto. Fica como aprendizado a importância de realizar um planejamento.

## 5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Por se tratar de um trabalho de conclusão de curso acadêmico, onde é necessário conciliar o tempo dedicado à sua realização com a demanda de outras disciplinas bem como com a atividade profissional, é natural olhar para trás e perceber atividades que poderiam ter sido feitas de melhor forma bem como aquelas que não puderam ser feitas conscientemente, em razão do tempo e da quantidade de recursos disponíveis para sua elaboração, ou ainda, identificar uma funcionalidade que ao ser implementada traria um maior valor agregado ao projeto, sem contar aquelas que porventura surgirem, sugeridas por pessoas que tiverem a oportunidade de ter contato com esse material. Fica aqui, uma relação, sem a menor pretensão de esgotar o assunto, de funcionalidades que ficam sugeridas para um trabalho futuro.

### 5.1.1 Aplicativo Móvel

Seria útil desenvolver um aplicativo para *smartphone*, que recebesse notificações de oportunidades de voluntariado publicadas, ou ainda notificações caso o usuário recebesse alguma mensagem ou solicitação de amizade. Nos dias de hoje, devido à dinâmica do dia a dia e ao fato de uma grande maioria da população utilizar

telefones com acesso à internet, isso contribuiria para aproximar ainda mais instituições e voluntários.

#### 5.1.2 Timeline

Por ser uma rede social voltada ao voluntariado, não são apresentadas apenas as publicações dos usuários do círculo de amizade. Dessa forma, o usuário pode perceber a necessidade de alguma instituição e se candidatar, tornar-se amigo, ou ainda comentar para algum outro usuário. Ainda assim, caberia criar uma configuração para que o usuário pudesse escolher se deseja visualizar somente as mensagens de seu círculo de amizade.

#### 5.1.3 Compartilhar Publicações

Essa funcionalidade foi retirada do *backlog* do produto por questões de tempo para análise e desenvolvimento da funcionalidade. Pelo fato da *timeline* exibir todas as publicações independente do círculo de amizade, essa funcionalidade foi deixada para trás, ficando sua implementação correlacionada com a implementação do item anterior.

#### 5.1.4 Cadastros

Não foram desenvolvidas interfaces para cadastramento de causas, habilidades e cidades, que devem ser mantidas através direto ao banco de dados. Essa funcionalidade seria interessante para o administrador da plataforma, caso percebesse alguma necessidade. Em especial, o cadastro de cidades e endereços poderia ser revisto para atender não só o mercado brasileiro, mas também outros países.

À medida que os voluntários realizarem trabalhos de voluntariado, pontuar a quantidade de horas em seus cadastrados e nos cadastros das instituições, criando um *ranking* de participação.

#### 5.1.5 Doações

A plataforma poderia ser melhorada com a possibilidade de o usuário realizar doações em espécie, através da utilização de cartão de crédito, geração de boleto bancário ou plataforma de pagamento online. Existem pessoas que não dispõem de tempo, mas dispõem de recursos e que, quando se sentissem motivadas, poderiam contribuir com alguma necessidade publicada pela instituição.

#### 5.1.6 Segurança

Na parte de segurança, não foi implementada a opção de enviar um link para aquele usuário que esquece a senha, permitindo-o realizar a alteração para poder acessar a plataforma novamente. Outra funcionalidade que poderia ser implementada, seria a autenticação de dois fatores e a funcionalidade de manter o usuário conectado, desobrigando-o de informar a senha para acesso, bem como a utilização de SSO (*Single Sign-On*) através de contas do *Facebook* ou *Google*, por exemplo.

#### 5.1.7 Veracidade do Certificado

Na emissão do certificado de trabalho voluntário, uma opção que poderia ser implementada é a impressão de um QR Code para que terceiros possam comprovar a veracidade do certificado. Além disso, incluir a utilização de certificado digital para que as instituições pudessem assiná-lo digitalmente, pois a assinatura digitalizada não possui valor legal para autenticação do certificado.

#### 5.1.8 Escalabilidade

O software foi desenvolvido utilizando um banco de dados relacional que não oferece escalabilidade para um grande número de usuários. O volume de informações no chat e na timeline seria melhor gerenciado se um banco de dados NoSQL fosse utilizado.

### 5.1.9 Oportunidades

As oportunidades publicadas poderiam ser repostadas automaticamente, de tempos em tempos como forma de incentivar os voluntários a se alistarem, aparecendo inclusive na *timeline* de voluntários que possuam compatibilidade de causa e habilidades requeridas.

Seria interessante também exibir as instituições que possuem mais participações de voluntários bem como as que possuem menos, como forma de balancear e incentivar os voluntários a procurar essas em vez daquelas.

Ainda sobre oportunidades, exibir os *cases* de sucesso, como forma de fazer propaganda e incentivar cada vez mais pessoas a participarem.

## REFERÊNCIAS

BASHAM, BRYAN; SIERRA, KATHY; BATES, BERT. **Use a Cabeça!** Servlets e JSP. Rio de Janeiro: Alta Books, p. 4,5, 37, 39, 54 e 283.

BLOG ROBSON CAMARGO PROJETOS E NEGÓCIOS. **O que são metodologias ágeis e quais as vantagens?** Não paginado. Disponível em: <<https://robsoncamargo.com.br/blog/O-que-sao-metodologias-ageis-e-quais-as-vantagens>>. Acesso em 11/10/2018.

BOOCH, GRADY. **UML: guia do usuário**. Elsevier: Rio de Janeiro, 2012. p. 14-31, 71-72,250.

BRASIL. Constituição (1988). **Constituição**: República Federativa do Brasil. Brasília, DF: Senado Federal, 1988.

BRASIL. Decreto 2.536, de 6 de Abril de 1998. Dispõe sobre a concessão do Certificado de Entidade de Fins Filantrópicos a que se refere o inciso IV do art. 18 da Lei nº 8.742, de 7 de dezembro de 1993, e dá outras providências. **Diário Oficial da União**, Brasília, DF, n. 66, 07 abr. 1998. Seção 1, p. 30.

BRASIL. Lei 91, de 28 de agosto de 1935. Determina regras pelas quaes são as sociedades declaradas de utilidade pública. **Diário Oficial da União**, Brasília, DF, n. 204, 4 set. 1935. p. 1.

BRASIL. Lei 9.608, de 18 de fevereiro de 1998. Dispões sobre o serviço voluntário e dá outras providências. **Diário Oficial da União**, Brasília, DF, n. 56, 19 fev. 1998. Seção 1, p. 94.

BRASIL. Lei 9.637, de 15 de maio de 1998. Dispõe sobre a qualificação de entidades como organizações sociais, a criação do Programa Nacional de Publicização, a extinção dos órgãos e entidades que menciona e a absorção de suas atividades por organizações sociais, e dá outras providências. **Diário Oficial da União**, Brasília, DF, n. 92, 18 mai. 1998. Seção 1, p. 20.

BRASIL. Lei 9.790, de 23 de março de 1999. Dispões sobre o serviço voluntário e dá outras providências. **Diário Oficial da União**, Brasília, DF, n. 35, 23 mar. 1999. Seção 1, p. 30.

CARDOSO, RUTH. Fortalecimento da Sociedade Civil In: IOSCHPE, EVELYN BERG. **3º Setor: desenvolvimento social sustentado**. Rio de Janeiro: Paz e Terra, 2005. p. 7.

CODD, E.F. **A Relational Model of Data for Large Shared Data Banks**. Communications of the ACM, Us, v. 13, n. 6, p. 377-387, jun. 1970.

COELHO, SIMONE DE CASTRO TAVARES. **Terceiro Setor: um estudo comparado entre Brasil e Estados Unidos**. 2. ed. São Paulo: SENAC, 2002. p. 39, nota 11.  
DAMIANI, EDGAR B. **Guia de Consulta Rápida JavaScript**. Novatec: São Paulo, 2006. p. 7.



CHIAVENATO, IDALBERTO. **Introdução à teoria geral da administração**. Barueri, SP: Manole, 2014. p. 24.

DIAS, MARIA TEREZA FONSECA. **Terceiro Setor e Estado**: legitimidade e regulação: por um novo marco jurídico. Belo Horizonte: Fórum, 2008. p. 97-128 e 219.  
FERNANDES, RUBEM CÉSAR. O que é o Terceiro Setor? In: IOSCHPE, EVELYN BERG. **3º Setor**: desenvolvimento social sustentado. Rio de Janeiro: Paz e Terra, 2005. p. 25-33.

FURGERY, SÉRGIO. **Java 2**: Ensino Didático: Desenvolvendo e Implementando Aplicações. Érica: São Paulo, 2002. p. 329, 355 e 357.

GIDDENS, ANTHONY. **Sociologia**. Porto Alegre: Penso, 2012. p. 578-579.  
FILANTROPIA. In: Dicionário Priberam da Língua Portuguesa, 2008-2013. Disponível em < <http://www.priberam.pt/dlpo/filantropia>>. Acesso em 25/09/2018.

GONCALVES, ANTONIO. **Beggining Java™ EE 6 Platform with Glassfish™ 3**. Apress: United States, 2010. p.34.

IBGE. **PNAD contínua em 2016 90,6% das mulheres e 74,1% dos homens realizaram afazeres domésticos ou cuidado de pessoas**. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/18566-pnad-continua-2016-90-6-das-mulheres-e-74-1-dos-homens-realizaram-afazeres-domesticos-ou-cuidados-de-pessoas>>. Acesso em 27/09/2018.

IBGE. **Voluntariado aumentou em 840 mil pessoas em 2017**. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/20913-voluntariado-aumentou-em-840-mil-pessoas-em-2017>>. Acesso em 17/11/2018.

INSTITUIÇÃO. In: MICHAELIS DICIONÁRIO BRASILEIRO DA LÍNGUA PORTUGUESA. Disponível em <<http://michaelis.uol.com.br/busca?r=0&f=0&t=0&palavra=instituição>>. Acesso em 25/09/2018.

IOSCHPE, EVELYN BERG. **3º Setor**: desenvolvimento social sustentado. Rio de Janeiro: Paz e Terra, 2005. Prefácio.

IRMANDADE DA SANTA CASA DE MISERICÓRDIA DE SÃO PAULO. Disponível em: < <http://www.santacasasp.org.br/portal/site>>. Acesso em 27/09/2018.

JANDL JUNIOR, PETER. **Java**: guia do programador: atualizado para Java 8. Novatec: São Paulo, 2015. p. 21-22, 89, 91, 128 E 148.

KALBACH, JAMES. **Design de navegação Web**. Porto Alegre: Bookman, 2009. p.206-207.

KANAAN, JOÃO CARLOS. **Informática Global**: tudo o que você precisa saber sobre informática. São Paulo: Pioneira, 1998. p. 177.

MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT. **Twelve Principles of Agile Software**. Curitiba, 2018 em: <<http://agilemanifesto.org/principles.html>>. Acesso: 11/10/2018.

MARTINS, JOSÉ CARLOS CORDEIRO. **Técnicas para gerenciamento de projetos de software**. Rio de Janeiro: Brasport, 2007. p. 1-12, 252-278.

MEC. **Escolas e universidades poderão contar horas de trabalho voluntário**. Disponível em: <<http://portal.mec.gov.br/ultimas-noticias/222-537011943/68251-escolas-e-universidades-poderao-contar-horas-de-trabalho-voluntario>>. Acesso: 17/11/2018.

MYSQL DOCUMENT STORE ARCHITECTURE. **MySQL Document Store**. Não paginado. Disponível em: <[https://www.mysql.com/products/enterprise/document\\_store.html](https://www.mysql.com/products/enterprise/document_store.html)>. Acesso em 21/10/2018.

ONU. **O trabalho voluntário e a ONU**. Disponível em: <<https://nacoesunidas.org/vagas/voluntariado/>>. Acesso em 07/10/2018.

RAMAKRISHANAN, RAGHU; GEHRKE, JOHANNES. **Sistemas de Banco de Dados**. 3. ed. São Paulo: MCGRAW-HILL, 2008. p. 3-10.

RAMALHO, JOSÉ ANTÔNIO. **Oracle 9i**. São Paulo: Berkeley Brasil, 2002. p. 8.  
RAMALHO, JOSÉ ANTÔNIO ALVES. **HTML Dinâmico**. São Paulo: Berkeley Brasil, 1999. p. 16.

SHAFFER, DAN. **HTML UTOPIA: Designing Without Tables Using CSS**. United States of America: Sitepoint, 2003. p. 14-16.

SAVOLA, TOM. **Usando HTML: O guia de referência mais completo**. Rio de Janeiro: Campus, 1997. p XXII.

SOUZA, MARCO AURÉLIO DE. **SQL, PL/SQL, SQL \*PLUS: manual de referência completo e objetivo**. Ciência Moderna Ltda: Rio de Janeiro, 2004. p. 29-30.

TELLES, ANDRÉ. **A Revolução das Mídias Sociais**. Cases, Conceitos, Dicas e Ferramentas. São Paulo: M. Books do Brasil Editora Ltda., 2011. p. 7.

THOMPSON, ANDRÉS A. Do Compromisso à eficiência? Os caminhos do terceiro setor na América Latina. In: IOSCHPE, EVELYN BERG. **3º Setor: desenvolvimento social sustentado**. Rio de Janeiro: Paz e Terra, 2005. p. 42.

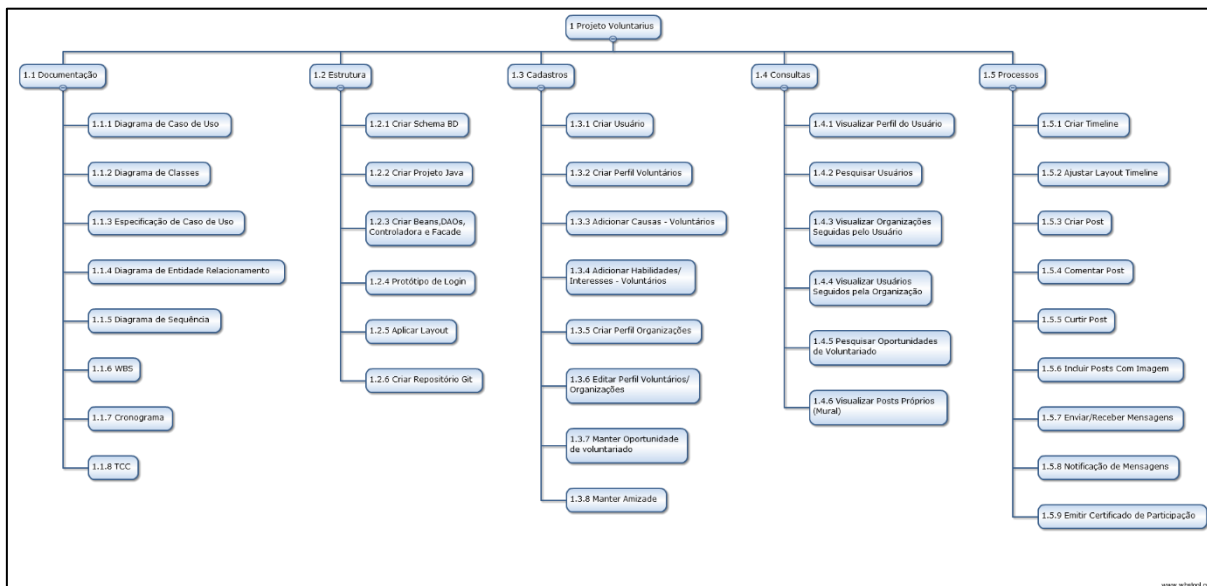
VARGAS, RICARDO VIANA. **Gerenciamento de Projetos: estabelecendo diferenciais competitivos**. Rio de Janeiro: Brasport, 2009. p. 174 e 175.

VILLELA, TATIANA. Em Movimento – 20 anos de Investimento Social no Brasil (legendado). **GIFE**, 21 out. 2015. Disponível em <<https://gife.org.br/quem-somos-gife/>>. Acesso em 25/09/2018.

WANDRESEN, RAFAEL ROMUALDO. **Orientação a Objetos II – Herança e Polimorfismo**. Curitiba, 04 de set. 2017. Aula proferida na Universidade Federal do Paraná.

## APÊNDICE A – ESTRUTURA ANALÍTICA DE PROJETO

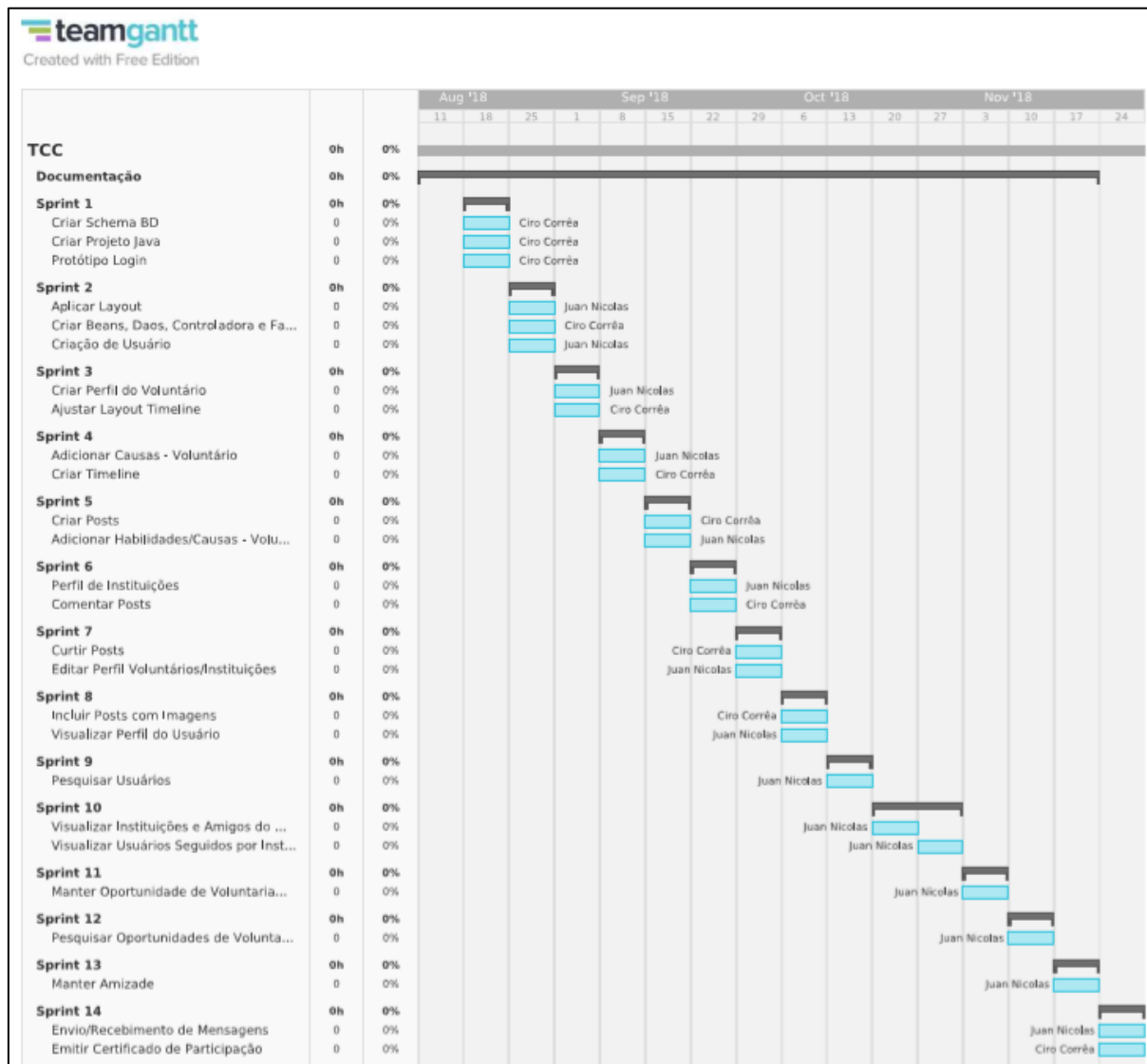
FIGURA 48 - ESTRUTURA ANALÍTICA DE PROJETO



FONTE: Os Autores (2018).

## APÊNDICE B – CRONOGRAMA DO PROJETO

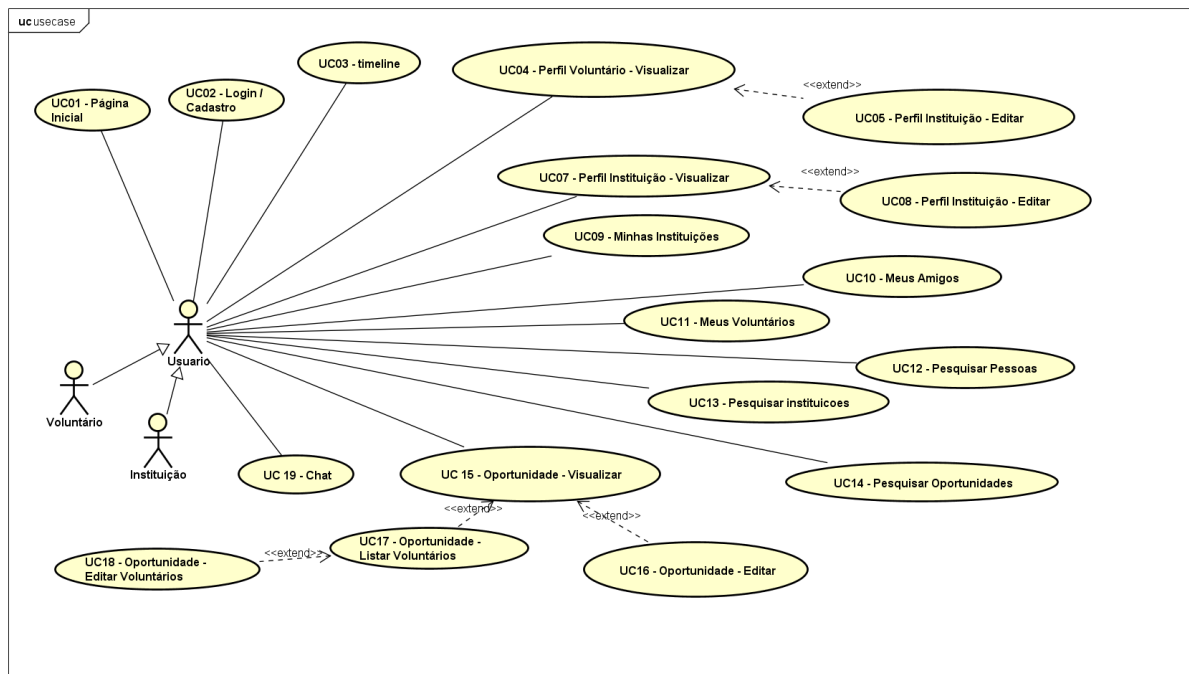
FIGURA 49 - CRONOGRAMA DO PROJETO



FONTE: Os Autores (2018).

## APÊNDICE C – DIAGRAMA DE CASO DE USO

FIGURA 50 - DIAGRAMA DE CASO DE USO



FONTE: Os Autores (2018).

## APÊNDICE D – ESPECIFICAÇÕES DE CASOS DE USO

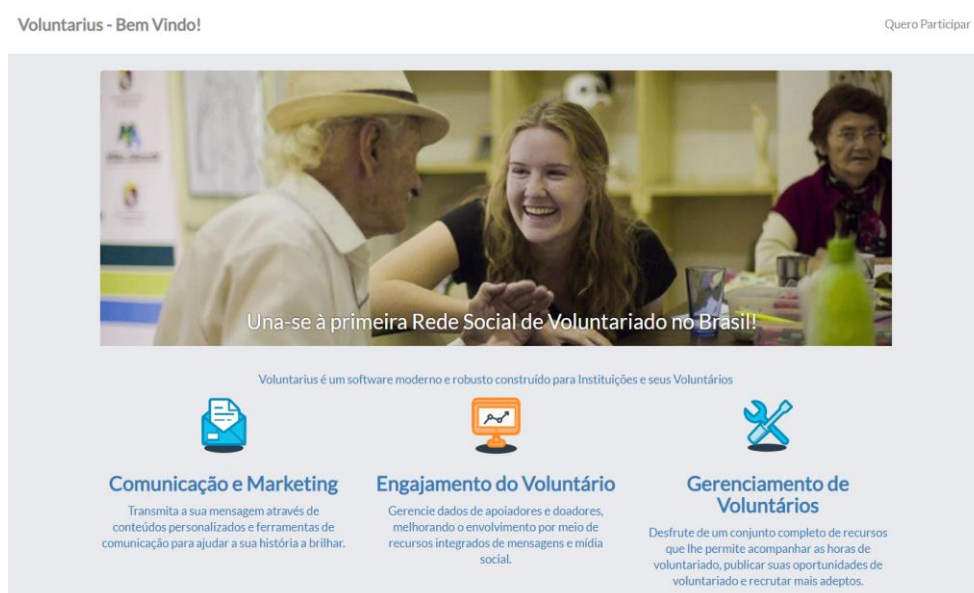
### UC01 – Página Inicial

#### Descrição

Este caso de uso serve para que o usuário conheça o site e para que possa participar.

#### Data View

#### DV01 - Tela de página inicial



FONTE: Os Autores (2018).

#### Ator Primário

Usuário.

#### Fluxo de Eventos Principal

1. A aplicação carrega e exibe a página inicial (DV01).
2. O usuário clica no botão “Quero Participar”.
3. A aplicação direciona para o caso de uso Login / Cadastro (UC02).
4. Fim de caso uso.

## UC02 – Login / Cadastro

### Descrição

Este caso de uso serve para que o usuário faça login ou se cadastre na aplicação.

### Data View

#### DV02 - Tela de login/cadastro

The image shows a web interface for user login and registration. It features two main sections side-by-side. The left section, titled 'Login', prompts the user to 'Acesse sua conta' and includes input fields for 'Endereço de Email' and 'Senha', followed by a blue 'Acessar' button. The right section, titled 'Cadastre-se', prompts the user to 'Crie uma nova conta' and includes input fields for 'Nome Completo', 'Endereço de Email', and 'Senha'. Below these fields is a checkbox labeled 'Quero cadastrar uma instituição' and a blue 'Cadastrar' button. The entire interface is overlaid on a background of a dense grid of numerous small, diverse profile pictures.

FONTE: Os Autores (2018).

### Pós-Condições

Após o fim normal deste caso de uso a aplicação deve:

1. Realizar o login do usuário, ou Incluir o cadastro do novo usuário.

### Ator Primário

Usuário.

### Fluxo de Eventos Principal

1. A aplicação carrega e exibe a interface de login / cadastro (DV02).



2. O usuário preenche o campo Endereço de E-mail, na seção “Login” **(R1) (R2) (A1)**.
3. O usuário preenche o campo Senha, na seção “Login” **(R1) (R3)**.
4. O usuário pressiona o botão “Acessar”.
5. A aplicação valida a combinação de e-mail e senha **(E1)**.
6. A aplicação direciona para o caso de uso UC05 - *Timeline*.
7. O caso de uso é encerrado.

### Fluxos Alternativos

**A1.** O usuário preenche a seção “Cadastre-se”.

1. O usuário preenche o campo Nome Completo, na seção “Cadastre-se” **(R1)**.
2. O usuário preenche o campo Endereço de e-mail, na seção “Cadastre-se” **(R1) (R2)**.
3. O usuário preenche o campo Senha, na seção “Cadastre-se” **(R1) (R3)**.
4. O usuário pressiona o botão “Cadastrar” **(A2)**.
5. A aplicação verifica se o e-mail do usuário já existe **(E2)**.
6. A aplicação mostra a notificação “Usuário cadastrado com sucesso!”.
7. O caso de uso é encerrado.

**A2.** O usuário clica no *check-box* “Quero cadastrar uma instituição”.

1. O usuário pressiona o botão “Cadastrar”.
2. A aplicação verifica se o e-mail do usuário já existe **(E2)**.
3. A aplicação mostra a notificação “Usuário cadastrado com sucesso!”.
4. O caso de uso é encerrado.

### Regras de Negócios

**R1.** Campo obrigatório.

**R2.** Valida formato de campo de e-mail, contendo arroba.

**R3.** Valida senha de acordo com política de senhas: mínimo 4 caracteres.

**R4.** Criptografia de senhas

1. A aplicação guarda as senhas no banco de forma criptografada.
2. Ao realizar o login, a aplicação não armazena a senha digitada, apenas compara-a com a senha armazenada no banco de dados.

**Fluxo de Exceção**

**E1.** A aplicação verifica que não existe a combinação de e-mail e senha

1. A aplicação apresenta a notificação “combinação de e-mail e senha não existe!”.
2. O caso de uso é finalizado.

**E2.** A aplicação verifica que já existe um usuário cadastrado com esse endereço de e-mail

1. A aplicação apresenta a notificação “Usuário já existe!”.
2. O caso de uso é finalizado.

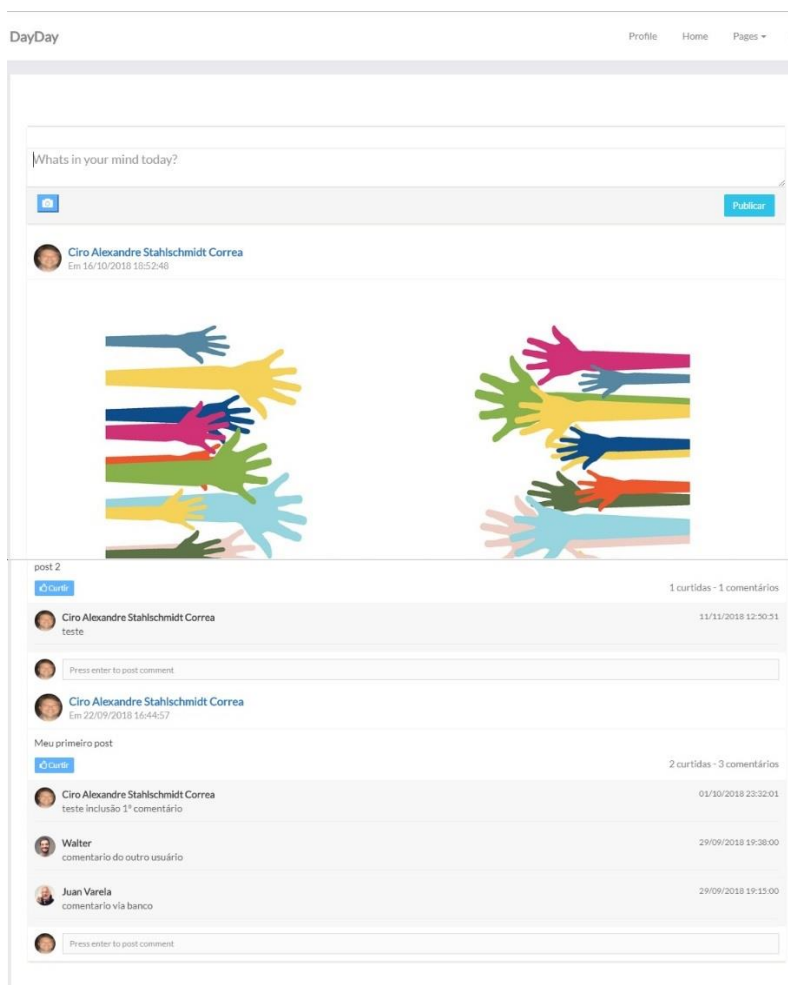
## UC03 – TimeLine

### Descrição

Este caso de uso serve para postar, mostrar, e interagir com os posts próprios e de amigos e organizações vinculados com o usuário.

### Data View

#### DV03 - Tela de *Timeline*



FONTE: Os Autores (2018).

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 - Login

### Pós-condições

Após o fim normal deste caso de uso a aplicação deve:

1. Ter salvo a postagem do usuário.

### Fluxo de Eventos Principal

1. A aplicação busca posts de pessoas e organizações vinculadas ao usuário **(R1)**.
2. A aplicação verifica se ativa a notificação de solicitações de amizade **(R2)**.
3. A aplicação verifica se ativa a notificação de chat **(R3)**.
4. A aplicação verifica se ativa a notificação de oportunidade **(R4)**.
5. A aplicação carrega a tela.
6. O usuário clica no campo “Qual é a boa?” **(A1) (A2)**.
7. A aplicação mostra o botão de upload de imagem, a combo de Visibilidade (Todos, Amigos), e o botão Post, embaixo desse campo.
8. O usuário preenche o campo “Qual é a boa?” com os dados do post.
9. O usuário pressiona o botão Post.
10. A aplicação grava o post.
11. A aplicação mostra o post como *label*.
12. O Caso de Uso é finalizado.

### Fluxo de Eventos Alternativo

**A1.** O usuário clica em Curtir, em algum post.

1. A aplicação grava ou apaga a curtida.
2. O Caso de Uso é finalizado.

**A2.** O usuário clica em Comentar, em algum post.

1. A aplicação mostra o botão de upload de imagem, o botão Cancelar, e o botão Compartilhar, embaixo do campo “faça algum comentário”.
2. O usuário preenche o campo “faça algum comentário” com os dados do post.
3. O usuário pressiona o botão Compartilhar.
4. A aplicação grava o post.
5. A aplicação mostra o post como *label*.
6. O Caso de Uso é finalizado.

**Regras de Negócio**

**R1.** Busca em ordem decrescente de Tempo, e filtrando por Relevância.

**R2.** Se a notificação de solicitações de amizade estiver ativa, aparece um ícone de um sino no menu.

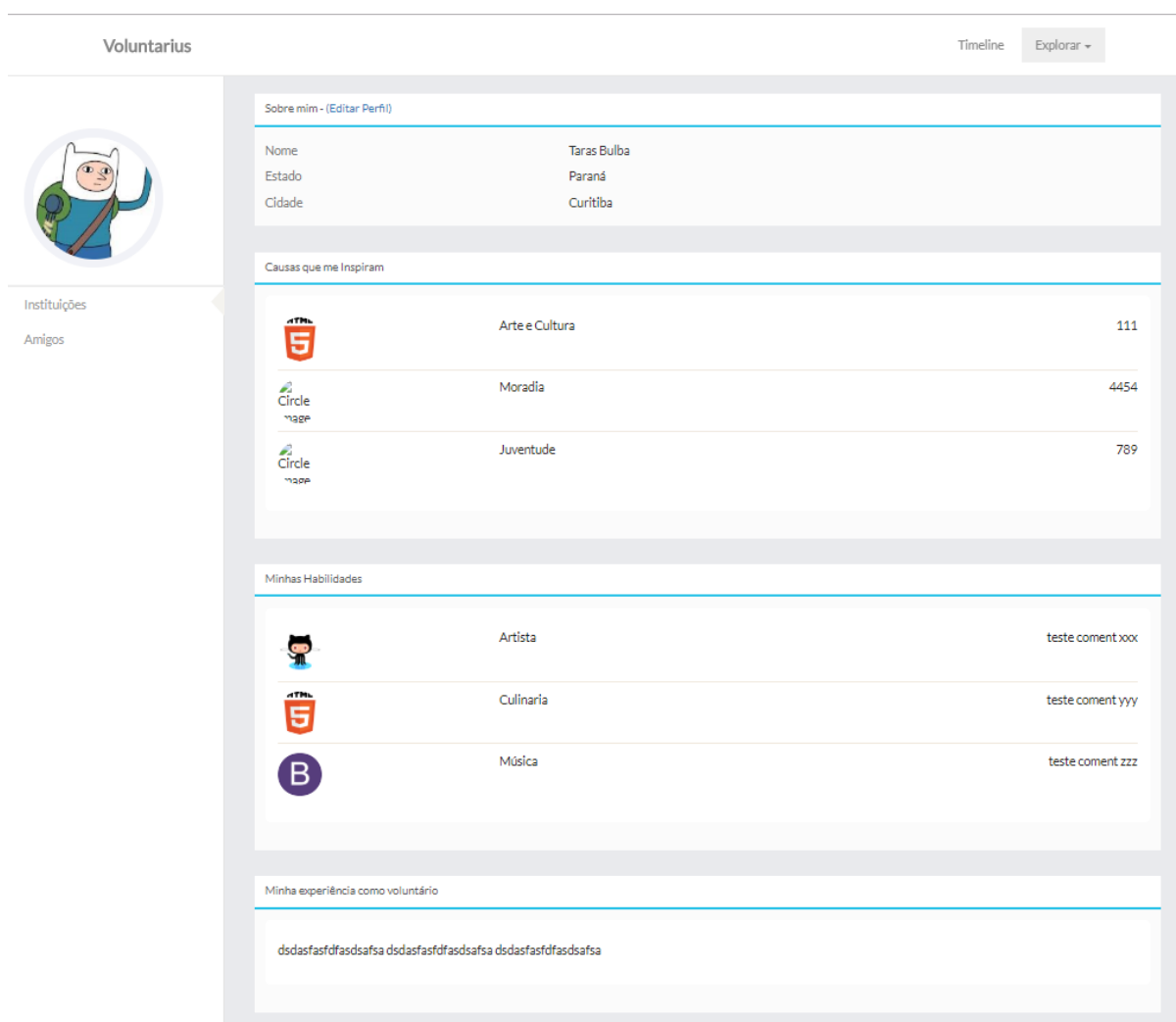
**R3.** Se a notificação de solicitações de amizade estiver ativa, aparece um ícone de mensagem no menu.

**R4.** Se a notificação de solicitações de voluntariado estiver ativa, aparece um ícone de mensagem no menu.

## UC04 – Perfil Voluntário - Visualizar

**Descrição**

Este caso de uso serve para visualizar o perfil do voluntário conectado, e também de outros voluntários.

**Data View****DV04 - Tela de Perfil Voluntário – Visualizar**

FONTE: Os Autores (2018).

**Pré-condições**

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.

### Pós-condições

Após o fim normal deste caso de uso a aplicação deve:

1. Exibir o perfil do usuário selecionado.

### Ator Primário

Usuário

### Fluxo de Eventos Principal

1. A aplicação verifica se o perfil do usuário é do usuário conectado.
2. A aplicação verifica se o perfil do usuário é de um amigo, ou se há solicitação de amizade pendente.
3. A aplicação carrega os dados do usuário selecionado **(R1)**.
4. A aplicação apresenta a tela do perfil do usuário selecionado.
5. O usuário clica no link “Editar Perfil” **(A1) (A2) (A3) (A4) (A5)**.
6. A aplicação direciona para o caso de uso UC05 (Perfil Voluntário - Editar);
7. O Caso de Uso é finalizado.

### Fluxos Alternativos

**A1.** O usuário clica no botão “Solicitar Amizade”.

1. A aplicação registra a solicitação para notificar o dono do perfil.
2. A aplicação substitui o botão por um *label* escrito “Solicitação Enviada” **(R1)**.
3. O Caso de Uso é finalizado.

**A2.** O usuário clica no botão “Desfazer Amizade”.

1. A aplicação exclui o vínculo de amizade.
2. A aplicação substitui o botão “Desfazer Amizade” pelo botão “Solicitar Amizade” **(R1)**.
3. O Caso de Uso é finalizado.

**A3.** O usuário clica no botão “Aceitar Amizade”.

1. A aplicação registra que os usuários são amigos.
2. A aplicação substitui o botão “Aceitar Amizade” pelo botão “Desfazer Amizade” **(R1)**.
3. O Caso de Uso é finalizado.

**A4.** O usuário clica no link “Instituições”.

1. A aplicação direciona para o caso de uso UC09 (Minhas Instituições).
2. O Caso de Uso é finalizado.

**A5.** O usuário clica no link “Amigos”.

1. A aplicação direciona para o caso de uso UC10 (Meus Amigos).
2. O Caso de Uso é finalizado.

### **Fluxos de Exceção**

Não possui.

### **Regras de Negócio**

**R1.** O perfil do usuário


1. Se for do usuário conectado: a aplicação exibe o link “Editar Perfil”.
2. Se for de outro usuário:
  - a. se não é amigo:
    - i. se ninguém solicitou amizade: a aplicação exibe o botão “Solicitar Amizade”;
    - ii. se eu já solicitei amizade: a aplicação exibe o *label* “Solicitação de amizade já foi realizada”;
    - iii. se ele já solicitou amizade: a aplicação exibe os botões “Aceitar Amizade” e “Rejeitar Amizade”;
  - b. se é amigo:
    - i. a aplicação exibe o botão “Desfazer Amizade”;
3. Ao clicar no botão “Solicitar Amizade” a aplicação registra uma notificação para o usuário solicitado, para que o mesmo autorize ou não a criação do vínculo de amizade. Nesse momento, o status da solicitação é P - Pendente.
4. Caso o usuário recuse a solicitação de amizade, o status da solicitação é alterado para R - Recusada.
5. Caso o usuário aceite a solicitação de amizade, o status da solicitação é alterado para C - Confirmada.



## UC05 – Perfil Voluntário - Editar

**Descrição:** Este caso de uso serve para que o voluntário possa atualizar seu perfil.

**Data View****DV05 - Tela de Perfil Voluntário – Editar**



Enviar Foto

Descrição de Causas e Habilidades

Sobre mim

Nome	Taras Bulba
Cpf	555888222
CEP	80050-530
Rua	João Dranka
Numero	66
Bairro	Cristo Rei
Telefone	(41) 99636-9536
UF	Paraná ▼
Cidade	Curitiba ▼

Causas que eu abraço

Arte e Cultura ▼	111
Moradia ▼	4454
Juventude ▼	789

Minhas Habilidades

Artista ▼	teste coment xoo
Culinaria ▼	teste coment yyy
Música ▼	teste coment zzz

Minha experiência como voluntário

dsdasfsofsofsa

dsdasfsofsofsa

dsdasfsofsofsa

Gravar Voltar

FONTE: Os Autores (2018).

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 - Login.
2. O dono do perfil for o usuário conectado.

### Pós-condições

Após o fim deste caso de uso, a aplicação deve:

1. Ter salvo as alterações realizadas pelo usuário.

### Fluxo de Eventos Principal

1. A aplicação busca os dados previamente gravados do perfil do voluntário.
2. A aplicação carrega a tela DV05 com os dados do usuário.
3. O usuário preenche o campo CPF **(R1)**.
4. O usuário preenche o campo CEP.
5. O usuário preenche o campo Rua.
6. O usuário preenche o campo Número.
7. O usuário preenche o campo Bairro.
8. O usuário preenche o campo Telefone.
9. O usuário seleciona a UF na combo UF.
10. A aplicação preenche a combo de Cidade, de acordo com a UF selecionada pelo usuário.
11. O usuário seleciona a cidade na combo Cidade.
12. O usuário preenche as combos de Causa da seção “Causas que eu abraço”.
13. O usuário preenche os campos de texto “Fale sobre sua causa” da seção “Causas que eu abraço”.
14. O usuário preenche os combos de Habilidades da seção “Minhas Habilidades”.
15. O usuário preenche os campos de texto “Fale sobre sua Habilidade” da seção “Minhas Habilidades”.
16. O usuário preenche o campo de texto “Fale um pouco sobre sua experiência” na seção “Minha experiência como voluntário”.
17. O usuário pressiona o botão “Gravar” **(A1) (A2) (A3)**.
18. A aplicação armazena os dados na tela.

- 19. A aplicação direciona para o caso de uso UC04 (Perfil Voluntário - Visualizar)
- 20. O Caso de Uso é finalizado.

### **Fluxo de Eventos Alternativo**

#### **A1. O usuário clica na foto do perfil**

- 1. A aplicação abre uma janela para que usuário selecione uma foto para fazer upload.
- 2. O usuário seleciona uma foto.
- 3. O usuário pressiona o botão “Enviar foto”.
- 4. A aplicação grava a foto enviada pelo usuário.
- 5. A aplicação direciona para o caso de uso UC04 (Perfil Voluntário - Visualizar)
- 6. O Caso de Uso é finalizado.

#### **A2. O usuário clica no link “Descrição de Causas e Habilidades”.**

- 1. A aplicação abre uma janela nova, com o caso de uso UC06 (Descrição de Causas e Habilidades).
- 2. O Caso de Uso é finalizado.

#### **A3. O usuário clica no botão “Voltar”.**

- 1. A aplicação direciona para o caso de uso UC04 (Perfil Voluntário - Visualizar).
- 2. O Caso de Uso é finalizado.

### **Regras de Negócio**

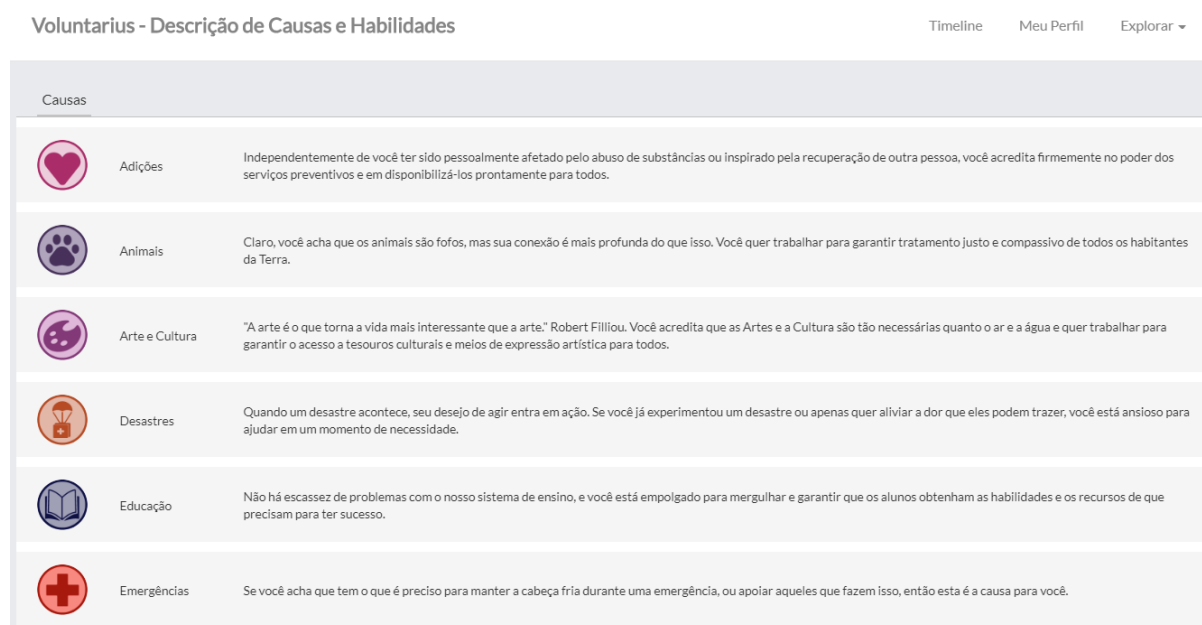
- R1.** O campo CPF é validado de acordo com algoritmo de validação de CPF.

## UC06 - Descrição de Causas e Habilidades

**Descrição:** Este caso de uso serve para que o usuário conheça a nomenclatura das causas e habilidades utilizadas na aplicação.

### Data View

#### DV06 - Tela de Descrição de Causas e Habilidades



FONTE: Os Autores (2018).

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 - Login.

### Fluxo de Eventos Principal

1. A aplicação carrega a tela DV06.
2. O Caso de Uso é finalizado.

## UC07 – Perfil Instituição - Visualizar

**Descrição:** Este caso de uso serve para visualizar o perfil da instituição conectada e também de outras instituições.

**Data View****DV07 - Tela de Perfil Instituição – Visualizar**


Voluntarius

Timeline Explorar ▾

Instituição - Detalhes - (Editar Perfil)

Nome	voluntarius
CNPJ	1234
Responsável	abcd
Endereço	Rua abcd nº 1234
CEP	1234
Estado	Paraná
Cidade	Londrina

Nossa Causa

 Animais

Quem Somos

abcd

FONTE: Os Autores (2018).

**Pré-condições**

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login

**Pós-condições**

Após o fim normal deste caso de uso a aplicação deve:

1. Exibir o perfil do usuário selecionado.

**Ator Primário**

Usuário

### Fluxo de Eventos Principal

1. A aplicação verifica se o perfil do usuário é do usuário conectado.
2. A aplicação carrega os dados do usuário selecionado **(R1)**.
3. A aplicação apresenta a tela do perfil DV07, do usuário selecionado.
4. O usuário clica no link “Editar Perfil” **(A1) (A2) (A3) (A4)**.
5. A aplicação direciona para o caso de uso UC08 (Perfil Instituição - Editar);
6. O Caso de Uso é finalizado.

### Fluxos Alternativos

**A1.** O usuário clica no botão “Seguir Instituição”.

1. A aplicação registra que o usuário está seguindo a instituição.
2. A aplicação substitui o botão “Seguir Instituição” pelo botão “Deixar de seguir Instituição” **(R1)**.
3. O Caso de Uso é finalizado.

**A2.** O usuário clica no botão “Deixar de seguir Instituição”.

1. A aplicação exclui o vínculo do voluntário com a instituição.
2. A aplicação substitui o botão “Deixar de seguir Instituição” pelo botão “Seguir Instituição” **(R1)**.
3. O Caso de Uso é finalizado.

**A3.** O usuário clica no link “Voluntários”.

1. A aplicação direciona para o caso de uso UC11 (Meus Voluntários).
2. O Caso de Uso é finalizado.

**A4.** O usuário clica no botão “Criar Oportunidade”.

1. A aplicação direciona para o caso de uso UC16 (Oportunidade - Editar).
2. O Caso de Uso é finalizado.

### Fluxos de Exceção

Não possui.

### Regras de Negócio

**R1.** O perfil do usuário:

1. Se for do usuário conectado: a aplicação exibe o link “Editar Perfil” e o botão “Criar Oportunidade”.


2. Se for de outro usuário:
  - a. se não está seguindo, a aplicação exibe o botão “Seguir Instituição”;
  - b. se está seguindo, a aplicação exibe o botão “Deixar de seguir Instituição”.

## UC08 – Perfil Instituição - Editar

**Descrição:** Este caso de uso serve para que a instituição possa atualizar seu perfil.

**Data View****DV08 - Tela de Perfil Instituição – Editar**

Voluntarius Timeline Explorar ▾



Enviar Foto

Descrição de Causas e Habilidades

Sobre a Instituição

Razao	voluntarius
CNPJ	1234
Responsável	abcd
CEP	1234
Rua	abcd
Numero	1234
Bairro	abcd
Telefone	1234
UF	Paraná ▾
Cidade	Londrina ▾

Nossa Causa

Animais ▾

Sobre Nós

abcd

Gravar Voltar

FONTE: Os Autores (2018).

**Pré-condições**

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 - Login.
2. O dono do perfil for o usuário conectado.



### **Pós-condições**

Após o fim deste caso de uso, a aplicação deve:

1. Ter salvo as alterações realizadas pelo usuário.

### **Fluxo de Eventos Principal**

1. A aplicação busca os dados previamente gravados do perfil da instituição.
2. A aplicação carrega a tela DV05 com os dados do usuário.
3. O usuário preenche o campo CNPJ **(R1)**.
4. O usuário preenche o campo Responsável.
5. O usuário preenche o campo CEP.
6. O usuário preenche o campo Rua.
7. O usuário preenche o campo Número.
8. O usuário preenche o campo Bairro.
9. O usuário preenche o campo Telefone.
10. O usuário seleciona a UF na combo UF.
11. A aplicação preenche a combo de Cidade, de acordo com a UF selecionada pelo usuário.
12. O usuário seleciona a cidade na combo Cidade.
13. O usuário preenche a combo de Causa da seção “Nossa Causa”.
14. O usuário preenche o campo de texto “Fale um pouco sobre sua instituição” na seção “Sobre nós”.
15. O usuário escolhe um arquivo contendo a assinatura digitalizada para impressão no certificado.
16. O usuário pressiona o botão “Gravar” **(A1) (A2) (A3)**.
17. A aplicação armazena os dados na tela.
18. A aplicação direciona para o caso de uso UC07 (Perfil Instituição - Visualizar)
19. O Caso de Uso é finalizado.

### **Fluxo de Eventos Alternativo**

**A1.** O usuário clica na foto do perfil

1. A aplicação abre uma janelar para que o usuário selecione uma foto para fazer upload.
2. O usuário seleciona uma foto.

3. O usuário pressiona o botão “Enviar foto”.
4. A aplicação grava a foto enviada pelo usuário.
5. A aplicação direciona para o caso de uso UC07 (Perfil Instituição - Visualizar)
6. O Caso de Uso é finalizado.

**A2.** O usuário clica no link “Descrição de Causas e Habilidades”.

1. A aplicação abre uma janela nova, com o caso de uso UC06 (Descrição de Causas e Habilidades).
2. O Caso de Uso é finalizado.

**A3.** O usuário clica no botão “Voltar”.

1. A aplicação direciona para o caso de uso UC07 (Perfil Instituição - Visualizar).
2. O Caso de Uso é finalizado.

### **Regras de Negócio**

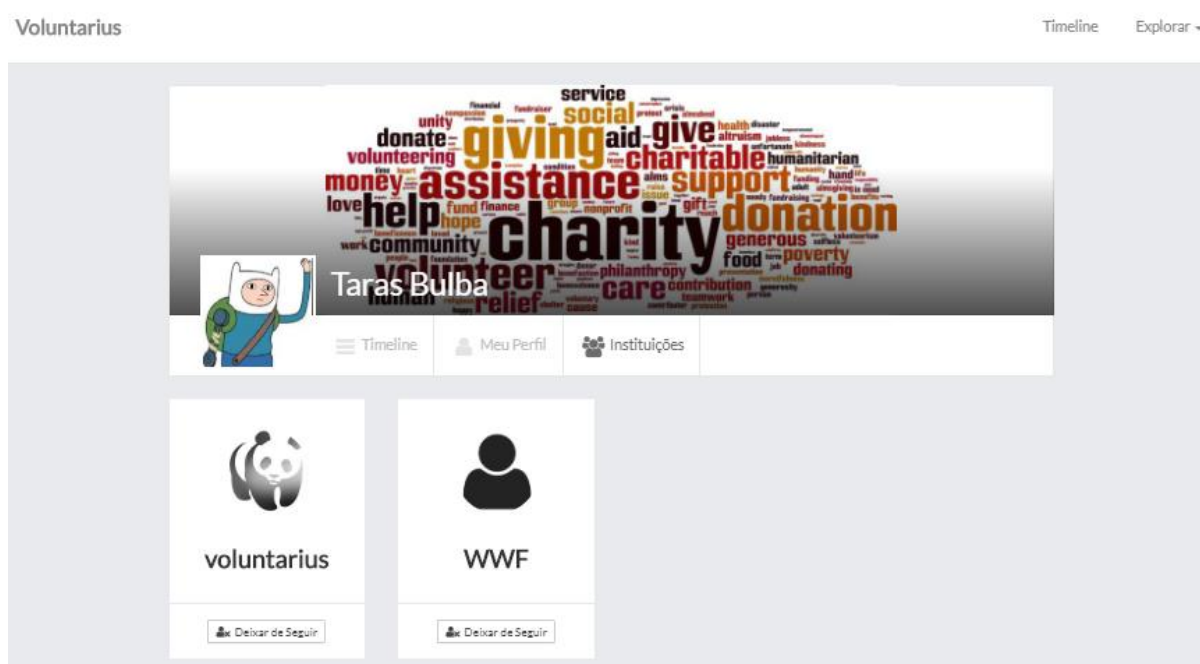
**R1.** O campo CNPJ é validado de acordo com algoritmo de validação de CNPJ.

## UC09 - Minhas Instituições

**Descrição:** Este caso de uso serve para visualizar quais instituições o usuário está seguindo.

**Data View**

## DV09 - Minhas Instituições.



FONTE: Os Autores (2018).

**Pré-condições**

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 - Login.

**Pós-condições.**

Após o fim normal deste caso de uso a aplicação deve:

1. Listar todas as instituições que o usuário está seguindo.

**Ator Primário**

Usuário.

**Fluxo de Eventos Principal**

1. A aplicação busca as instituições que o usuário está seguindo.
2. A aplicação carrega a tela DV09.
3. A aplicação apresenta a lista de instituições **(A1) (A2)**.
4. O caso de uso é finalizado.

**Fluxo Alternativo**

**A1.** Usuário clica em uma instituição.

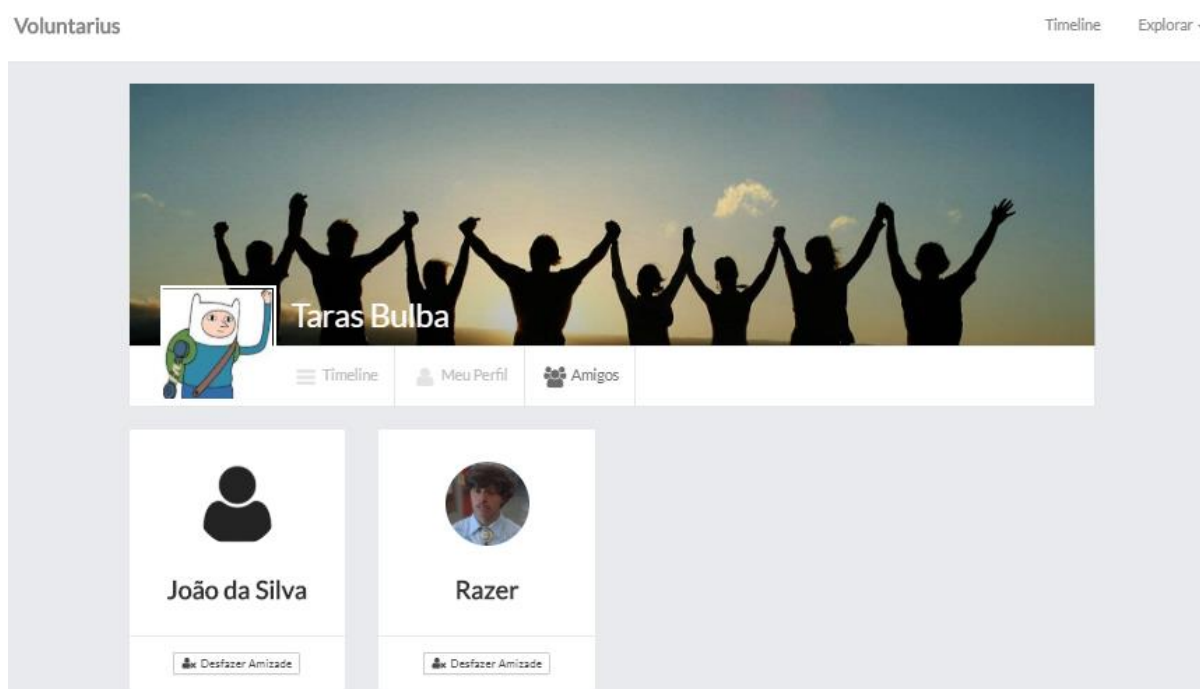
1. O usuário clica em um determinado voluntário.
2. A aplicação carrega o UC07 (Perfil Instituição - Visualizar).
3. O Caso de Uso é finalizado.

**A2.** Usuário clica no botão “Deixar de seguir”.

1. A aplicação exclui o vínculo do voluntário com a instituição.
2. A aplicação recarrega a tela DV09.
3. O Caso de Uso é finalizado.

## UC10 - Meus Amigos

**Descrição:** Este caso de uso serve para visualizar os amigos do voluntário.

**Data View****DV10 - Meus Amigos.**

FONTE: Os Autores (2018).

**Pré-condições**

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 - Login.

**Pós-condições**

Após o fim normal deste caso de uso a aplicação deve:

1. Listar todos os voluntários com os quais o usuário tem amizade.

**Ator Primário**

Usuário.

**Fluxo de Eventos Principal**

1. A aplicação busca os voluntários com os quais o usuário tem amizade.
2. A aplicação carrega a tela.
3. A aplicação lista os voluntários.
4. O Caso de Uso é finalizado **(A1) (A2) (A3) (A4)**.

**Fluxo Alternativo**

**A1.** Usuário clica em um voluntário.

1. O usuário clica em um determinado voluntário.
2. A aplicação carrega o UC04 (Perfil Voluntário - Visualizar).
3. O Caso de Uso é finalizado.

**A2.** Usuário clica no botão “Iniciar Chat”.

1. A aplicação carrega o UC19 (Chat).
2. O Caso de Uso é finalizado.

**A3.** Usuário clica no botão “Aceitar Amizade”.

1. A aplicação atualiza o vínculo de amizade com aquele usuário.
2. A aplicação recarrega a tela DV10.
3. O Caso de Uso é finalizado.

**A4.** Usuário clica no botão “Rejeitar Amizade”.

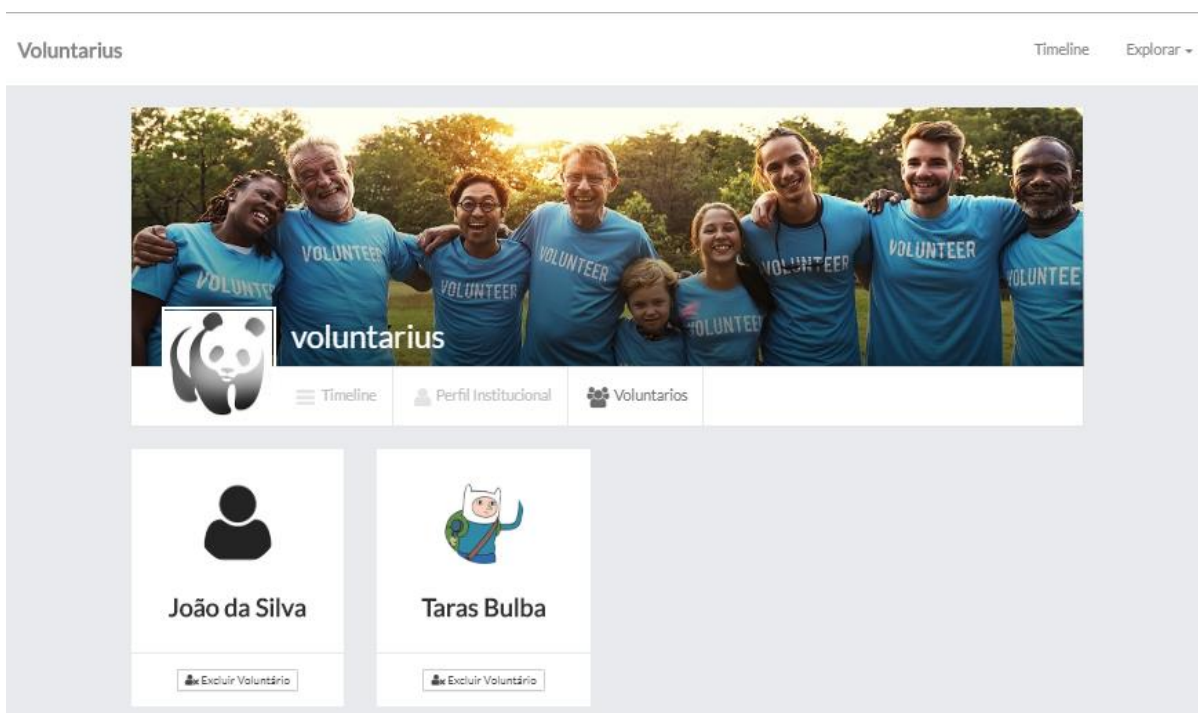
1. A aplicação atualiza o vínculo de amizade com aquele usuário.
2. A aplicação recarrega a tela DV10.
3. O Caso de Uso é finalizado.

## UC11 - Meus Voluntários

**Descrição:** Este caso de uso serve para visualizar os voluntários que estão seguindo a instituição.

### Data View

#### DV11 - Meus Voluntários.



FONTE: Os Autores (2018).

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 - Login.

### Pós-condições.

Após o fim normal deste caso de uso a aplicação deve:

1. Listar todos os voluntários que estão seguindo a instituição.

**Ator Primário**

Usuário.

**Fluxo de Eventos Principal**

1. A aplicação busca os voluntários que estão seguindo a instituição.
2. A aplicação carrega a tela DV11.
3. O Caso de Uso é finalizado **(A1) (A2) (A3)**.

**Fluxo Alternativo**

**A1.** Usuário clica em um voluntário.

1. O usuário clica em um determinado voluntário.
2. A aplicação carrega o UC04 (Perfil Voluntário - Visualizar).
3. O Caso de Uso é finalizado.

**A2.** Usuário clica no botão “Excluir Voluntário”.

1. A aplicação exclui o vínculo de amizade com aquele usuário.
2. A aplicação recarrega a tela DV11.
3. O Caso de Uso é finalizado.

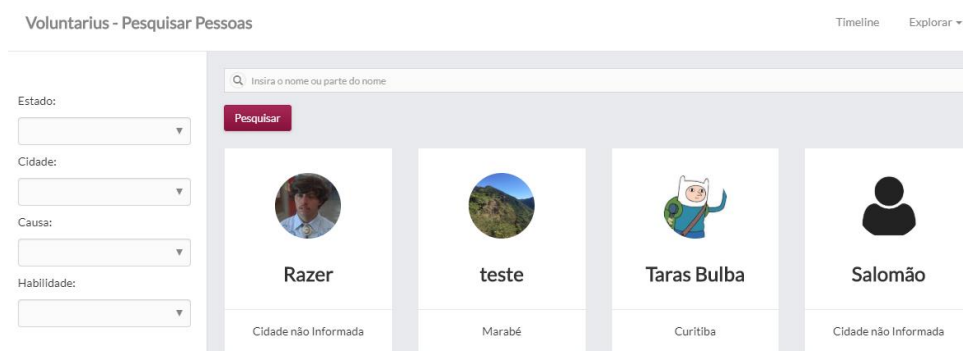


## UC12 – Pesquisar Pessoas

**Descrição:** Este caso de uso serve para pesquisar voluntários.

### Data View

#### DV12 - Tela de Pesquisar Pessoas.



### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login

### Pós-condições

Após o fim normal deste caso de uso a aplicação deve:

1. listar os usuários conforme o critério de pesquisa.

### Ator Primário

Usuário

### Fluxo de Eventos Principal

1. A aplicação carrega as combos de estado, causa e habilidade.
2. O usuário digita o nome no campo de pesquisa (lupa) **(R1)**.
3. O usuário seleciona um estado, na combo Estado **(R1)**.
4. A aplicação carrega a combo de cidade, conforme o estado selecionado pelo usuário.
5. O usuário seleciona uma cidade, na combo Cidade **(R1)**.

6. O usuário seleciona uma causa, na combo Causa **(R1)**.
7. O usuário seleciona uma habilidade, na combo habilidade **(R1)**.
8. O usuário pressiona o botão “Pesquisar”.
9. A aplicação exibe os registros encontrados **(E1)**.
10. O Caso de Uso é finalizado **(A1)**.

### **Fluxos Alternativos**

**A1.** Nome do usuário clicado.

1. O usuário clica no nome do usuário encontrado.
2. A aplicação executa o caso de uso UC04 (Perfil Voluntário - Visualizar).
3. O Caso de Uso é finalizado.

### **Fluxos de Exceção**

**E1.** Nenhum registro encontrado:

1. A aplicação exibe uma mensagem informando que não encontrou nenhum usuário contendo o nome digitado.
2. O Caso de Uso é finalizado.

### **Regras de Negócio**

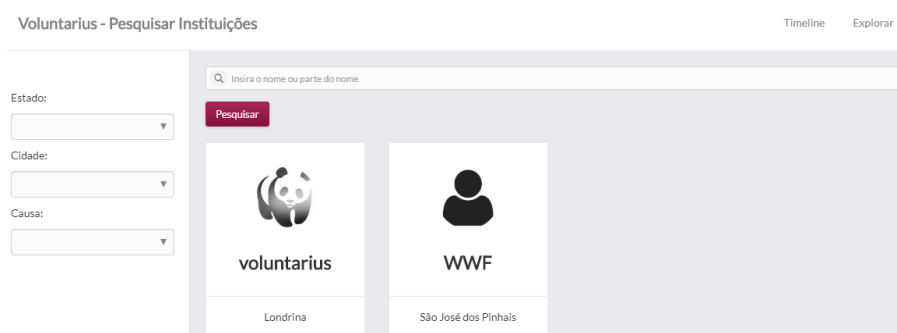
**R1.** Não obrigatório

## UC13 – Pesquisar Instituições

**Descrição:** Este caso de uso serve para pesquisar instituições.

### Data View

#### DV13 - Tela de Pesquisar Instituições.



FONTE: Os Autores (2018).

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login

### Pós-condições

Após o fim normal deste caso de uso a aplicação deve:

1. Listar os usuários conforme o critério de pesquisa.

### Ator Primário

Usuário

### Fluxo de Eventos Principal

1. A aplicação carrega as combos de estado, e causa.
2. O usuário digita o nome no campo de pesquisa (lupa) **(R1)**.
3. O usuário seleciona um estado, na combo Estado **(R1)**.
4. A aplicação carrega a combo de cidade, conforme o estado selecionado pelo usuário.
5. O usuário seleciona uma cidade, na combo Cidade **(R1)**.

6. O usuário seleciona uma causa, na combo Causa **(R1)**.
7. O usuário pressiona o botão “Pesquisar”.
8. A aplicação exibe os registros encontrados **(E1)**.
9. O Caso de Uso é finalizado **(A1)**.

### **Fluxos Alternativos**

**A1.** Nome do usuário clicado.

1. O usuário clica no nome do usuário encontrado.
2. A aplicação executa o caso de uso UC07 (Perfil Instituição - Visualizar).
3. O Caso de Uso é finalizado.

### **Fluxos de Exceção**

**E1.** Nenhum registro encontrado:

1. A aplicação exibe uma mensagem informando que não encontrou nenhum usuário contendo o nome digitado.
2. O Caso de Uso é finalizado.

### **Regras de Negócio**

**R1.** Não obrigatório

## UC14 – Pesquisar Oportunidades

**Descrição:** Este caso de uso serve para pesquisar oportunidades.

### Data View

#### DV14 - Tela de Pesquisar Oportunidades.

FONTE: Os Autores (2018).

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login

### Pós-condições

Após o fim normal deste caso de uso a aplicação deve:

1. Listar as oportunidades de voluntariado conforme o critério de pesquisa.

### Ator Primário

Usuário

### Fluxo de Eventos Principal

1. A aplicação carrega as combos de estado, causa, e habilidade.
2. O usuário digita o nome no campo de pesquisa (lupa) **(R1)**.
3. O usuário seleciona um estado, na combo Estado **(R1)**.

4. A aplicação carrega a combo de cidade, conforme o estado selecionado pelo usuário.
5. O usuário seleciona uma cidade, na combo Cidade **(R1)**.
6. O usuário seleciona uma causa, na combo Causa **(R1)**.
7. O usuário seleciona uma habilidade, na combo Habilidade **(R1)**.
8. O usuário seleciona uma opção do *radio-button* Presencial **(R1)**.
9. O usuário seleciona uma opção no *check-box* Minhas Oportunidades **(R2)**.
10. O usuário pressiona o botão “Pesquisar”.
11. A aplicação exibe os registros encontrados, de acordo com os critérios de pesquisa, ordenados por data de cadastro, em ordem do mais recente para o mais antigo **(E1)**.
12. O Caso de Uso é finalizado **(A1)**.

### Fluxos Alternativos

**A1.** Oportunidade clicada.

1. O usuário clica em uma oportunidade de voluntariado encontrada.
2. A aplicação executa o caso de uso UC15 (Oportunidade - Visualizar).
3. O Caso de Uso é finalizado.

### Fluxos de Exceção

**E1.** Nenhum registro encontrado:

1. A aplicação exibe uma mensagem informando que não encontrou nenhum usuário contendo o nome digitado.
2. O Caso de Uso é finalizado.

### Regras de Negócio

**R1.** Não obrigatório.

**R2.** Ao selecionar “minhas oportunidades”, a pesquisa restringe os resultados para que apareçam somente aquelas que o usuário está vinculado. Ou seja, se é um voluntário, aparecem aquelas nas quais ele solicitou participar, apresentando uma notificação caso tenha sido aceito. Já se for uma instituição, aparecem aquelas que ela criou e publicou, apresentando uma notificação caso algum voluntário tenha solicitado participar.

## UC15 – Oportunidade - Visualizar

**Descrição**

Este caso de uso serve para visualizar as oportunidades de voluntariado.

**Data View**


**DV15** - Tela de Oportunidade - Visualizar.

Voluntarius - Oportunidade - Visualizar Timeline Explorar ▾

---

Criar Oportunidade




Descrição	teste
Presencial	Sim



Data de Início	16/11/2018
Data de Finalização	16/11/2018
Vagas	3
Carga Horária	4
Status	Publicada

---

Habilidade(s) Requerida(s)

	Artista
	Música
	Matemática

Editar Oportunidade
Listar Voluntarios

FONTE: Os Autores (2018).

**Pré-condições**

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login

## Ator Primário

Instituição

## Fluxo de Eventos Principal

1. A aplicação busca os dados da oportunidade de voluntariado.
2. A aplicação verifica se o dono da oportunidade é o usuário conectado.
3. A aplicação apresenta a tela (DV15) **(R1)**.
4. O usuário clica no botão “Editar Oportunidade” **(A1) (A2) (A3)**.
5. A aplicação direciona para o caso de uso UC16 (Oportunidade - Editar).
6. O caso de uso é finalizado.

## Fluxos Alternativos

**A1:** Botão “Listar Voluntários” pressionado.

1. A aplicação direciona para o caso de uso UC17 (Oportunidade - Listar Voluntários).
2. Encerra o caso de uso.

**A2:** Botão “Quero Participar” pressionado.

1. A aplicação registra a solicitação do voluntário.
2. A aplicação substitui o botão “Quero Participar” por um *label* informando que já foi solicitado, está sendo processado.
3. Encerra o caso de uso.

**A3:** Botão “Imprimir Certificado” pressionado.

1. A aplicação busca os dados do voluntário.
2. A aplicação busca os dados da instituição.
3. A aplicação busca os dados da oportunidade.
4. A aplicação gera o certificado com os dados obtidos.
5. O certificado é exibido na tela do usuário.
6. Encerra o caso de uso.

## Regras de Negócio

R1. A aplicação verifica se o usuário é dono da oportunidade:

1. Se sim:



- a. a aplicação exibe o botão Listar Voluntários.
  - b. a aplicação verifica se a oportunidade foi Finalizada:
    - i. se não: a aplicação exibe o botão Editar Oportunidade.
2. Se não, a aplicação verifica se o usuário está vinculado à oportunidade:
- a. se não, aparece o botão “Quero Participar”.
  - b. se sim, a aplicação verifica o status do voluntário:
    - i. se ainda não foi aceito/rejeitado, mostra label informando sua solicitação ainda não foi processada.
    - ii. se foi aceito, mostra label informando que já foi aceito.
    - iii. se foi rejeitado, mostra label informando que não há mais vagas.
    - iv. se foi aceito, e já concluiu sua participação, mostra label agradecendo, e exibe botão “Imprimir Certificado”.

## UC16 – Oportunidade - Editar


**Descrição:** Este caso de uso serve para editar as oportunidades de voluntariado.

**Data View****DV16 - Tela de Oportunidade - Editar.**

Voluntarius - Oportunidade - Editar Timeline Explorar ▾

---

Criar Oportunidade

Descrição	<input type="text" value="teste"/>
Presencial	<input checked="" type="radio"/> Sim <input type="radio"/> Não
	
Data de Início	<input type="text" value="16/11/2018"/>
Data de Finalização	<input type="text" value="16/11/2018"/>
Vagas	<input type="text" value="3"/>
Carga Horária	<input type="text" value="4"/>
Status	<input style="border: 1px solid #ccc;" type="text" value="Publicada"/>
Habilidade 1	<input style="border: 1px solid #ccc;" type="text" value="Artista"/>
Habilidade 2	<input style="border: 1px solid #ccc;" type="text" value="Música"/>
Habilidade 3	<input style="border: 1px solid #ccc;" type="text" value="Matemática"/>

FONTE: Os Autores (2018).

**Pré-condições**

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login
2. O usuário for dono da oportunidade.

**Pós-condições**

Após o fim normal deste caso de uso a aplicação deve:

1. Ter salvo os dados da oportunidade de voluntariado.

## **Ator Primário**

Instituição

## **Fluxo de Eventos Principal**

1. A aplicação carrega o campo Status **(R1)**.
2. A aplicação carrega as combos de habilidades.
3. A aplicação busca os dados gravados da oportunidade.
4. A aplicação apresenta a tela (DV16).
5. A instituição preenche o campo Descrição, com um título para a oportunidade.
6. A instituição informa se a(s) atividade(s) necessita(m) da presença pessoal na instituição ou não.
7. A instituição escolhe uma imagem para representação da oportunidade de voluntariado.
8. A instituição informa a data de início e término da oportunidade de voluntariado **(E1) (R2) (R5)**.
9. A instituição digita o número de vagas disponíveis **(R3)**.
10. A instituição informa a carga horária total para fins de informação ao voluntário bem com geração do certificado de participação.
11. A instituição pode informar as habilidades necessárias para a realização do voluntariado.
12. A instituição digita uma descrição, explicando em linhas gerais a(s) atividades a serem realizadas na oportunidade de voluntariado.
13. A instituição clica no botão “Gravar” **(A1) (A2) (E1) (E2)**.
14. A aplicação armazena no banco os dados da tela.
15. A aplicação direciona para o caso de uso UC15 (Oportunidade - Visualizar).
16. O caso de uso é finalizado.

## **Fluxos Alternativos**

**A1:** Botão “Voltar” pressionado.

1. A aplicação direciona para o caso de uso UC15 (Oportunidade - Visualizar).

2. Encerra o caso de uso.

### **Fluxos de Exceção**

**E1.** Campos pendentes de preenchimento:

1. A aplicação consiste se o campo foi preenchido.
2. A aplicação destaca o campo obrigatório.
3. Encerra o caso de uso.

**E2.** Edição de campos da oportunidade, após o campo Status estar como “Publicada”.

1. A aplicação exibe uma mensagem informando que a oportunidade já foi publicada, e agora só pode ser Finalizada.
2. Encerra o caso de uso.

### **Regras de Negócio**

**R1.** Uma oportunidade de voluntário possui as seguintes situações:

1. E – Editada: não consta nas pesquisas, somente a instituição dona pode enxergar. Neste momento, é possível fazer alterações em qualquer campo.
2. P – Publicada: momento em que a oportunidade se torna visível para os voluntários. Não permite mais editar, somente finalizar.
3. F – Finalizada: oportunidade encerrada. Não consta nas pesquisas, somente a instituição dona pode enxergar. Não permite mais editar.

**R2.** A data de término deve ser igual ou maior que a data de início.

**R3.** O número de vagas deve ser maior ou igual a 1.





## UC17 – Oportunidade - Listar Voluntários

**Descrição:** Este caso de uso serve para visualizar os voluntários vinculados a uma oportunidade.

### Data View

#### DV17 - Tela de Oportunidade - Listar Voluntários.

Voluntarius Timeline Explorar ▾

Voluntário	Início	Término	Status	Horas	Editar
 Taras Bulba - Curitiba/PR	21/11/2018	21/11/2018	Aceito	6	
 teste - Marabá/PA			Rejeitado	0	

FONTE: Os Autores (2018).

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login.
2. O usuário for uma instituição dona da oportunidade.

### Ator Primário

Instituição

### Fluxo de Eventos Principal

1. A aplicação busca os voluntários vinculados à oportunidade de voluntariado.
2. A aplicação apresenta a tela (DV17).
3. O usuário clica no link do lápis na coluna “Editar” **(A1) (A2)**.
4. A aplicação direciona para o caso de uso UC18 (Oportunidade - Editar Voluntário).
5. O caso de uso é finalizado.

**Fluxos Alternativos**

**A1:** Nome do voluntário pressionado.

1. A aplicação direciona para o caso de uso UC04 (Perfil Voluntário - Visualizar).
2. Encerra o caso de uso.

**A2:** Botão “Voltar” pressionado.


1. A aplicação direciona para o caso de uso UC15 (Oportunidade - Visualizar).
2. Encerra o caso de uso.

## UC18 – Oportunidade - Editar Voluntário

**Descrição:** Este caso de uso serve para visualizar os voluntários vinculados a uma oportunidade.

### Data View

#### DV18 - Tela de Oportunidade - Editar Voluntário.

Voluntário	Início	Término	Status	Horas
 <b>Taras Bulba</b> <small>- Curitiba/PR</small>	21/11/2018	21/11/2018	Aceitar ▼	6

Gravar

FONTE: Os Autores (2018).

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login
2. O usuário for uma instituição dona da oportunidade.

### Ator Primário

Instituição

### Fluxo de Eventos Principal

1. A aplicação busca os dados do vínculo do voluntário com a oportunidade.
2. A aplicação apresenta a tela (DV18).
3. O usuário preenche a data de início.
4. O usuário preenche a data de término **(R1)**.
5. O usuário preenche o status.
6. O usuário preenche as Horas **(R1)**.
7. O usuário clica no botão Gravar **(A1)**.
8. A aplicação armazena os dados da tela no banco.

9. A aplicação direciona para o caso de uso UC17 (Oportunidade - Listar Voluntários).
10. O caso de uso é finalizado.

### **Fluxos Alternativos**

**A1:** Botão “Voltar” pressionado.

1. A aplicação direciona para o caso de uso UC17 (Oportunidade - Listar Voluntários).
2. O caso de uso é finalizado.

### **Regras de Negócio**

**R1.** O preenchimento deste campo é requisito para que o voluntário possa enxergar o botão “Imprimir Certificado” no caso de uso UC15 (Oportunidade - Visualizar).

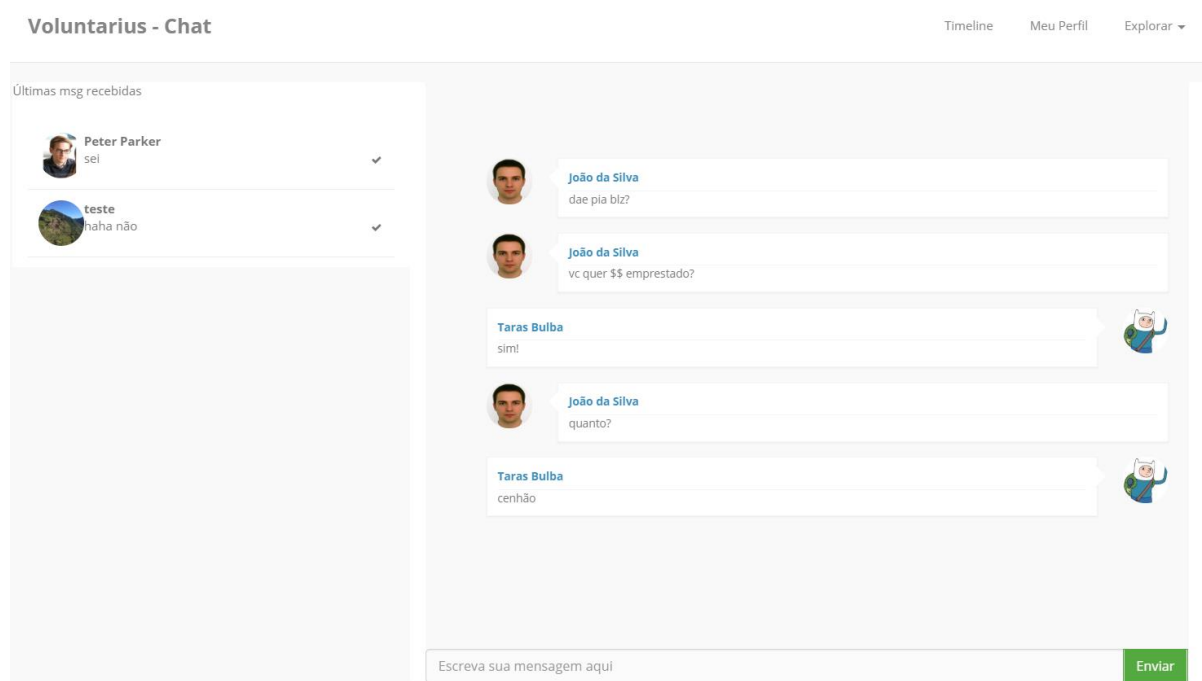


## UC19 – Chat

**Descrição:** Este caso de uso serve para que os voluntários possam conversar via chat.

### Data View

#### DV19 - Tela de Chat.



FONTE: Os Autores (2018).

### Pré-condições

Este caso de uso pode iniciar somente se:

1. A aplicação tiver executado o UC02 – Login

### Ator Primário

Instituição

### Fluxo de Eventos Principal

1. A aplicação busca todas as conversas que o usuário ainda não respondeu, em ordem decrescente de data.

2. A aplicação busca as vinte últimas mensagens com o usuário selecionado para realizar o chat, em ordem decrescente de data.
3. A aplicação seta a conversa atual como “lida”.(R1)
4. A aplicação apresenta a tela (DV19).
5. O usuário preenche o campo “Escreva sua mensagem aqui”.(A1)
6. O usuário pressiona o botão “Enviar”.
7. O caso de uso é finalizado.

### **Fluxos Alternativos**

**A1:** O usuário clica em uma conversa do lado esquerdo da tela.

1. Este caso de uso é reiniciado desde o passo 1, considerando o usuário da conversa selecionada.
2. O caso de uso é finalizado.

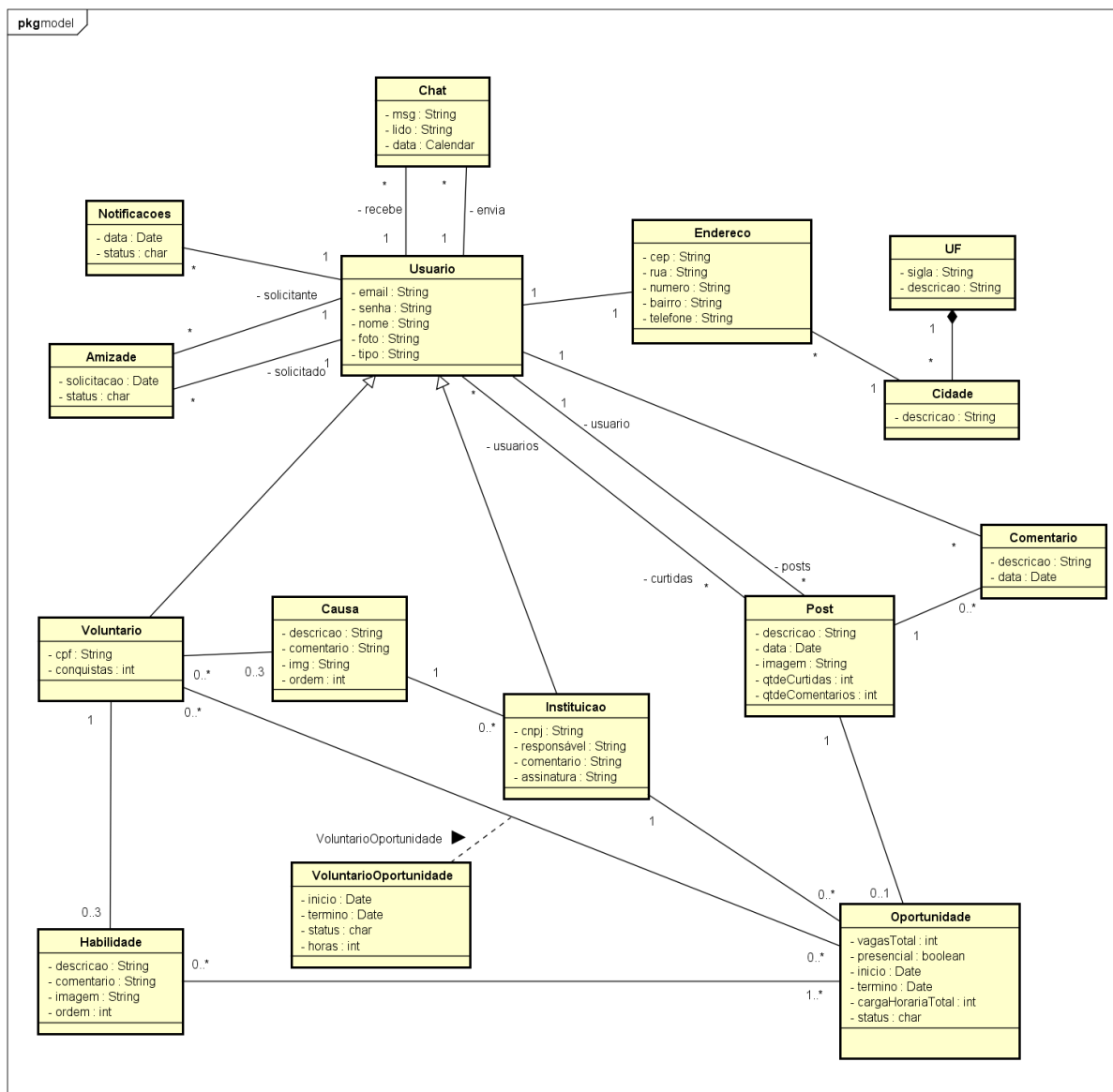
### **Regras de Negócio**

**R1.** Flag Lido:

1. Se uma conversa não foi lida, isso dispara a notificação na conversa, no lado esquerdo da tela. Se já foi lida, não aparecem mais notificações.

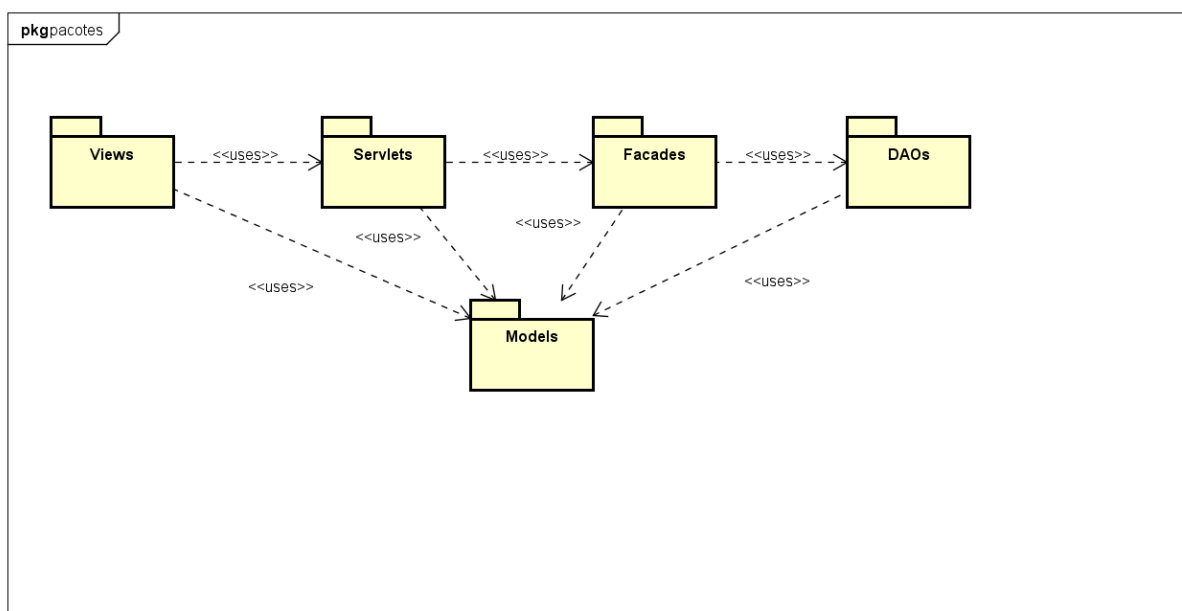
## APÊNDICE E – DIAGRAMA DE CLASSES

FIGURA 51 - DIAGRAMA DE CLASSES



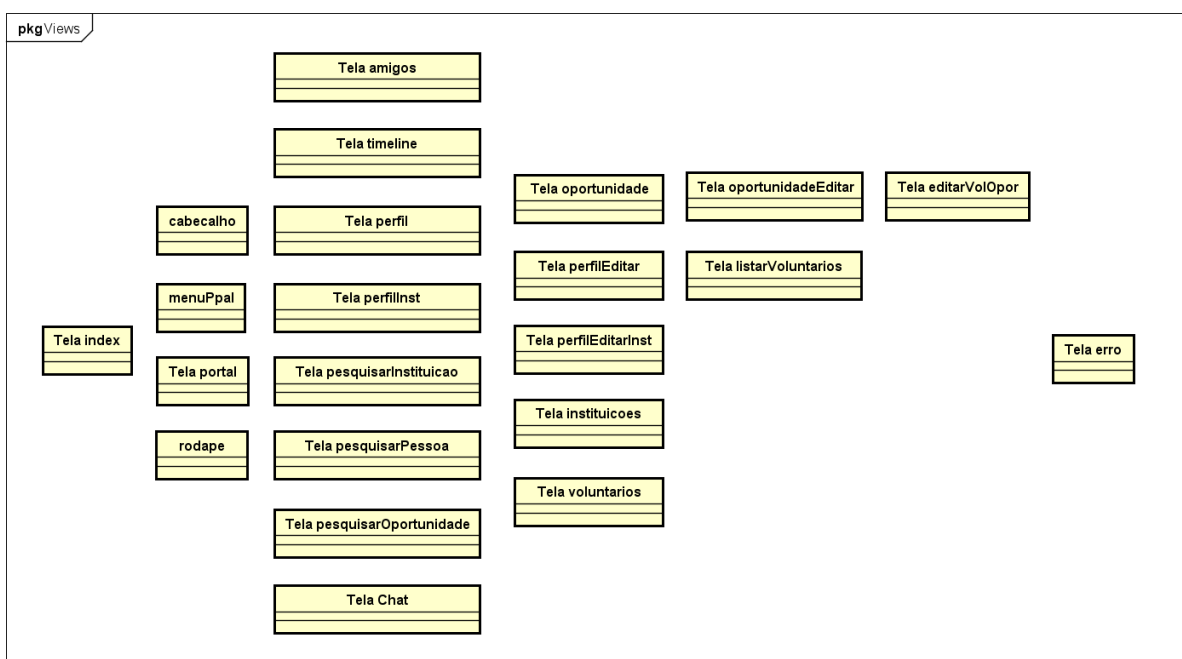
FONTE: Os Autores (2018).

FIGURA 52 – DIAGRAMA DE CLASSE PACOTES



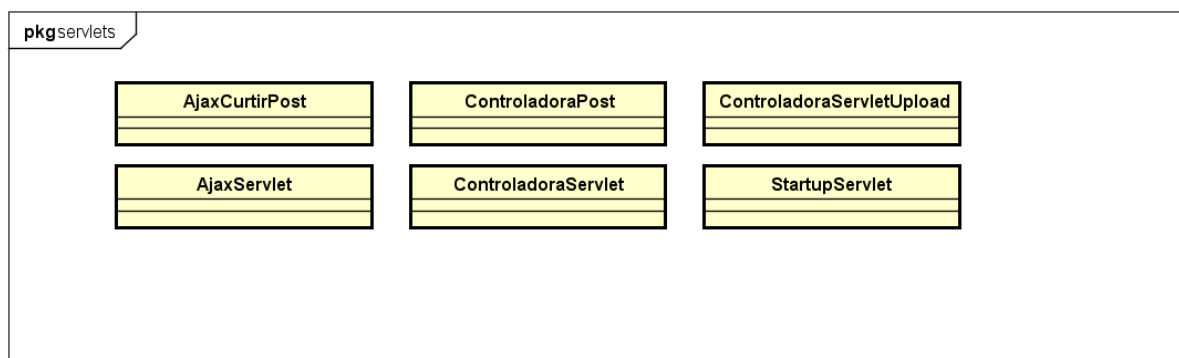
FONTE: Os Autores (2018).

FIGURA 53 – DIAGRAMA DE CLASSE VIEWS



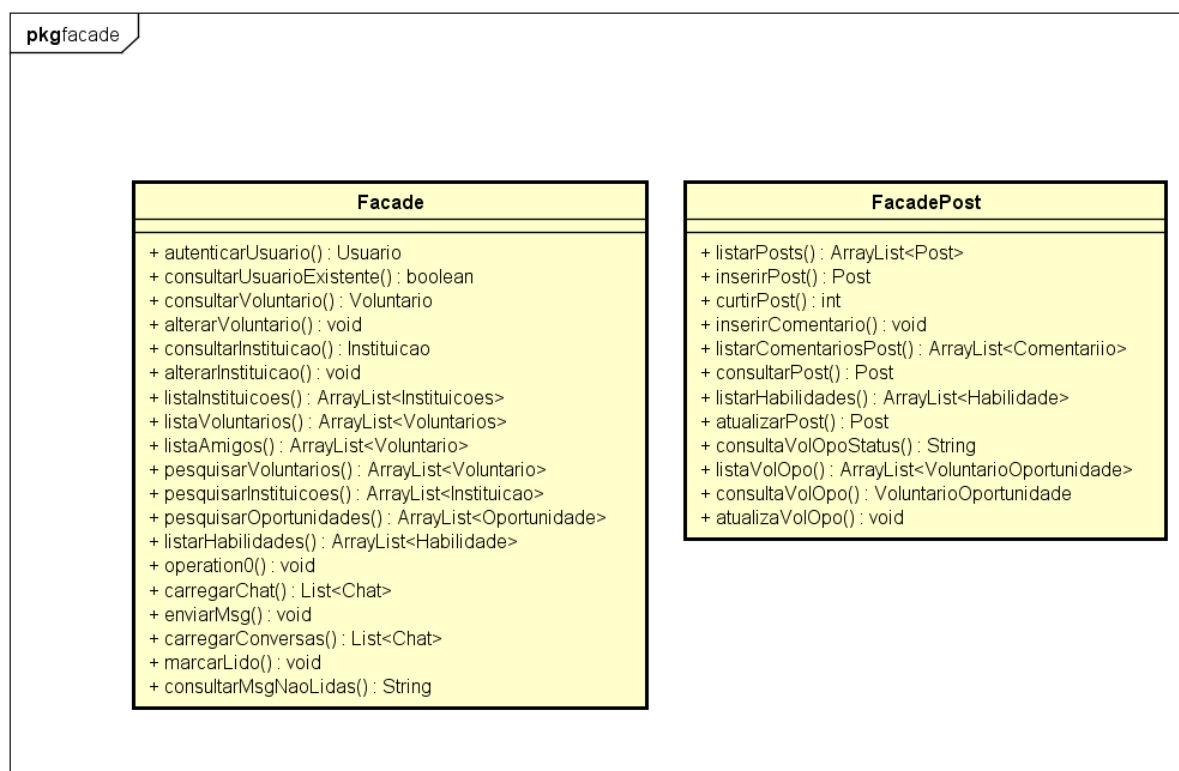
FONTE: Os Autores (2018).

FIGURA 54 – DIAGRAMA DE CLASSE CONTROLLER



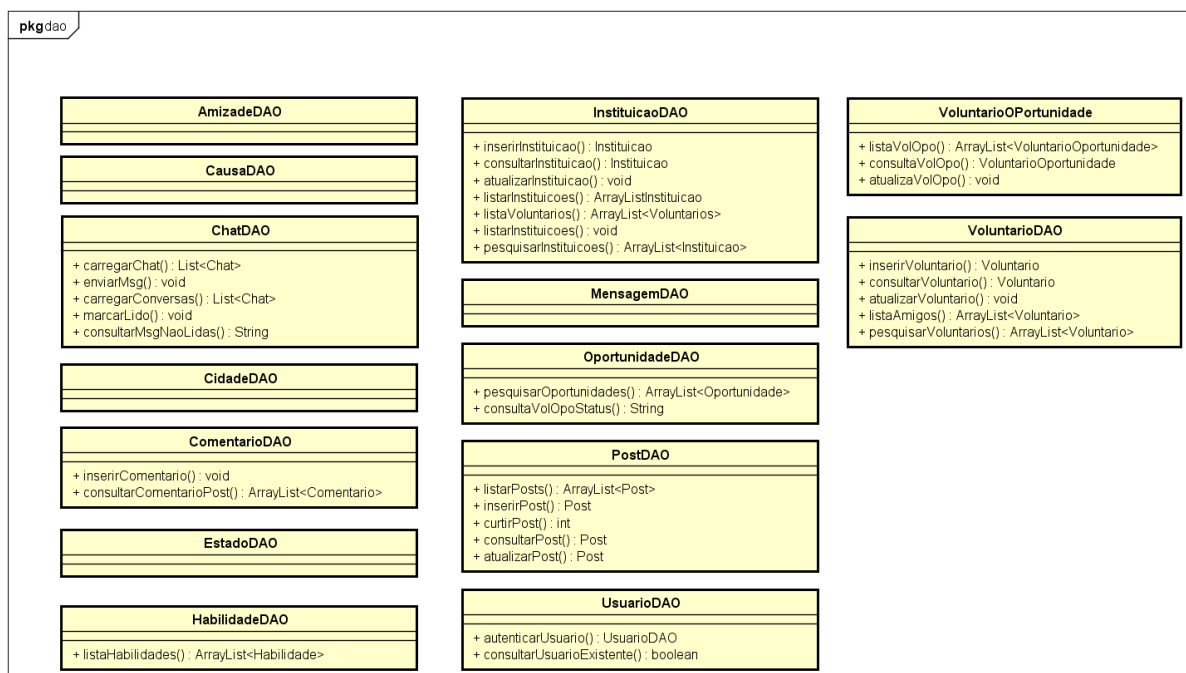
FONTE: Os Autores (2018).

FIGURA 55 – DIAGRAMA DE CLASSE FAÇADE



FONTE: Os Autores (2018).

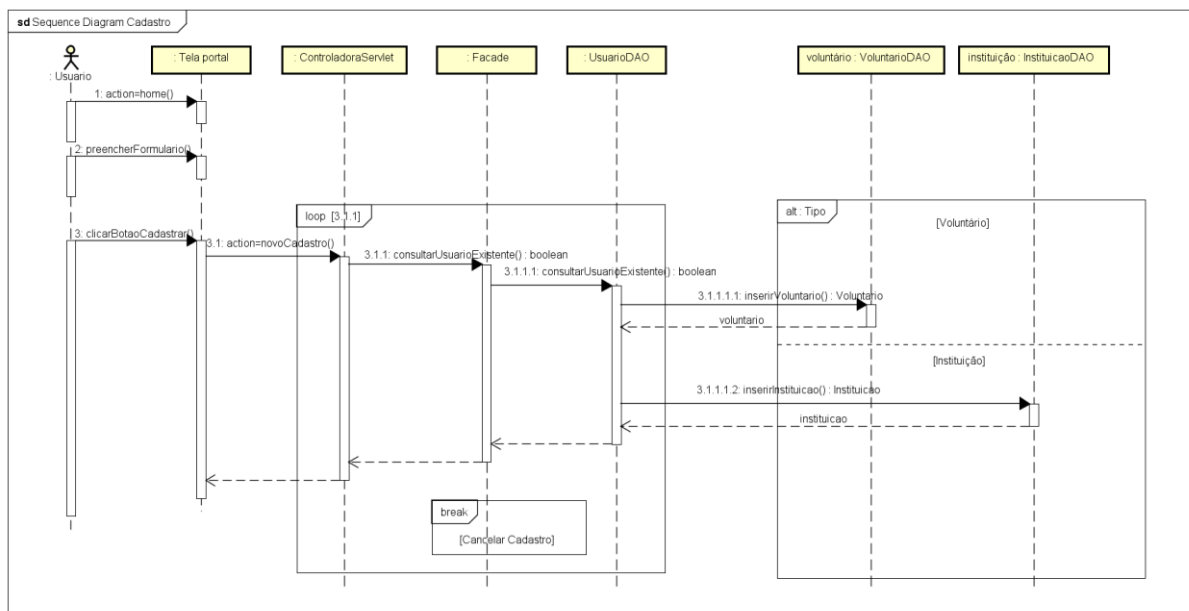
FIGURA 56 – DIAGRAMA DE CLASSE DAO



FONTE: Os Autores (2018).

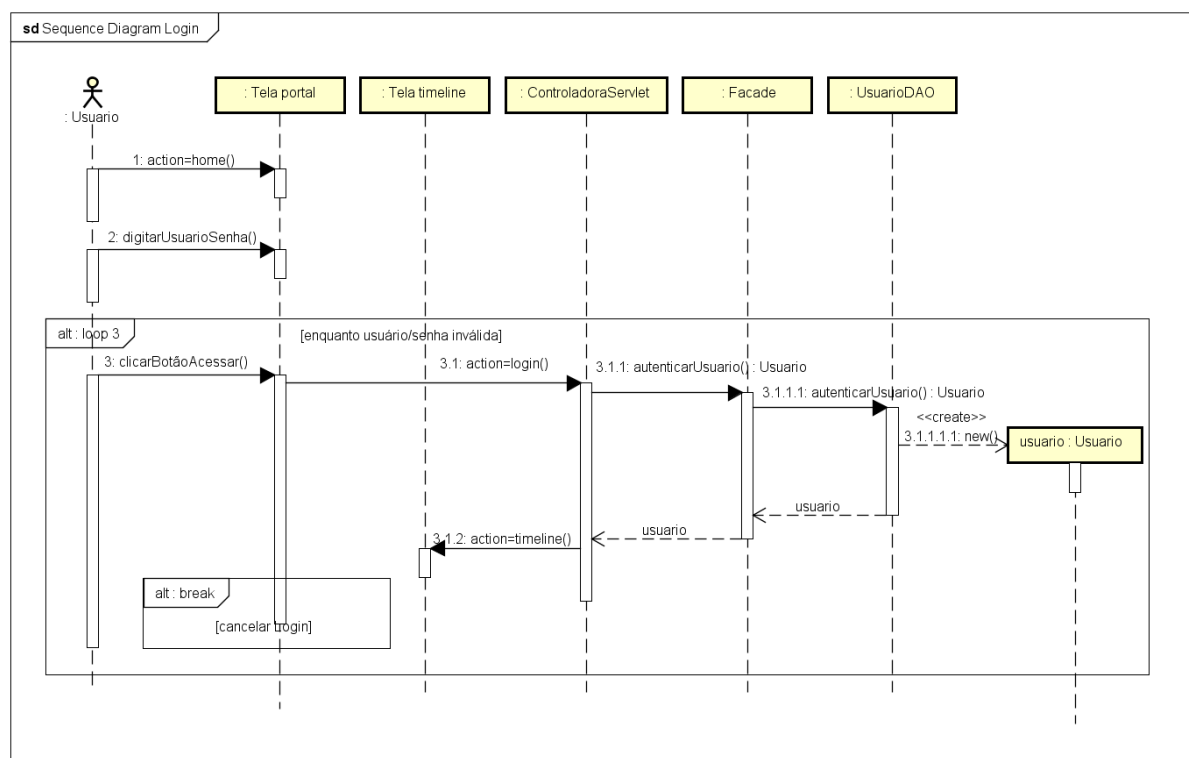
## APÊNDICE F – DIAGRAMAS DE SEQUÊNCIA

FIGURA 57 - EFETUAR CADASTRO



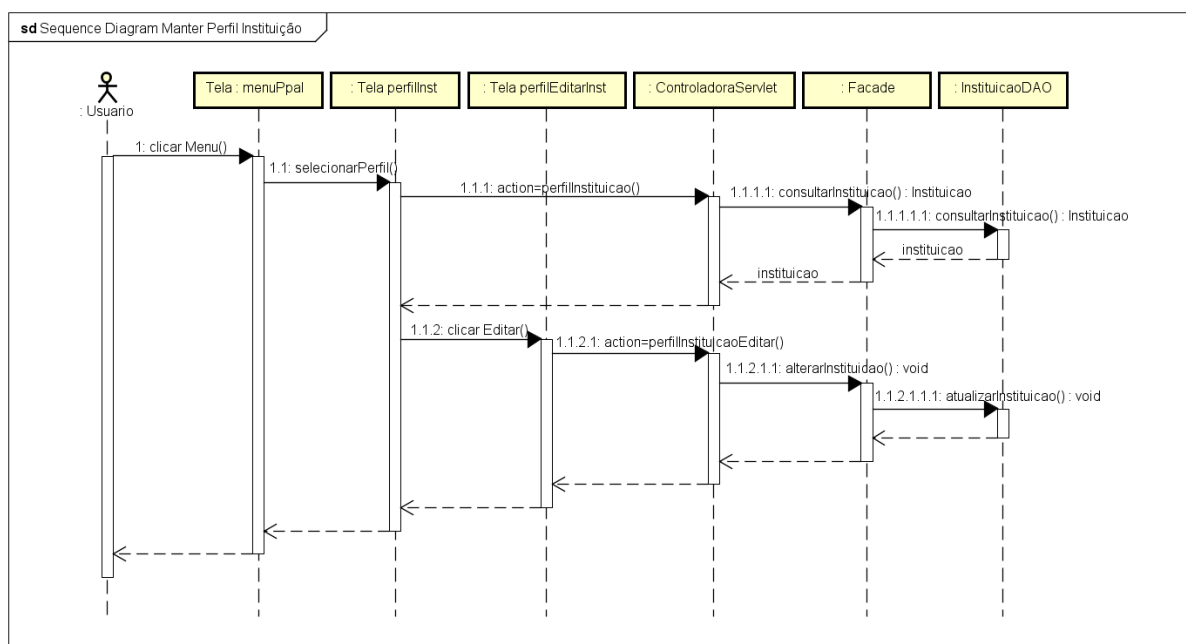
FONTE: Os Autores (2018).

FIGURA 58 - EFETUAR ACESSO



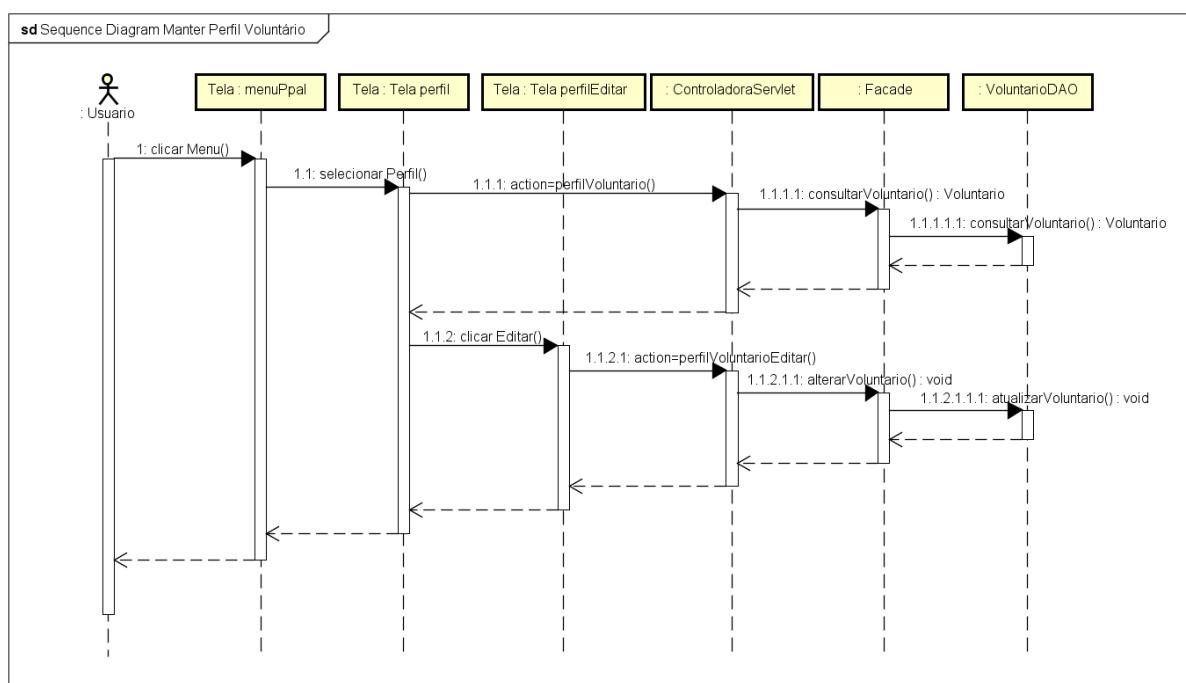
FONTE: Os Autores (2018).

FIGURA 59 - MANTER PERFIL INSTITUIÇÃO



FONTE: Os Autores (2018).

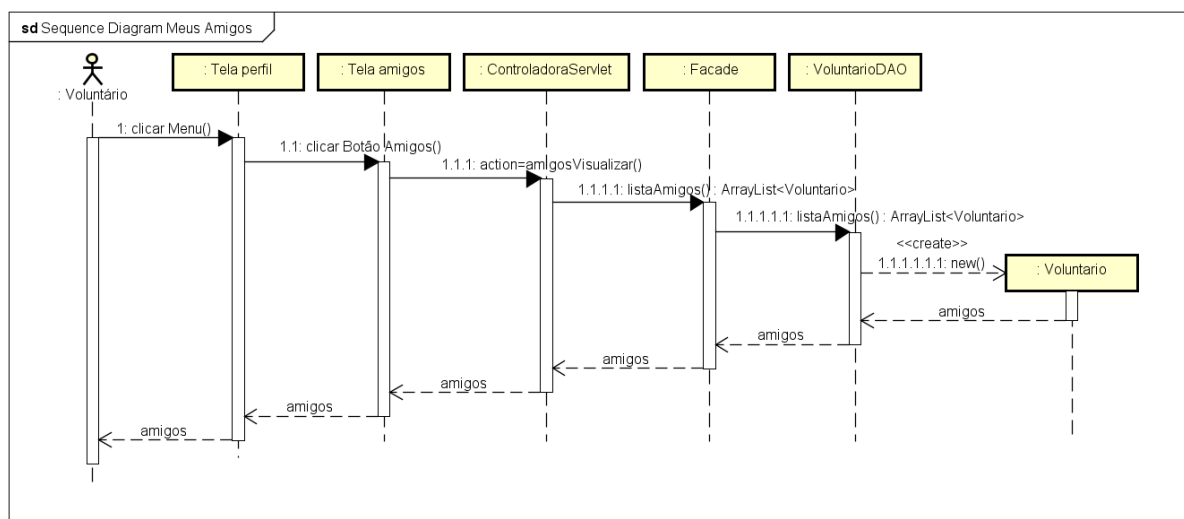
FIGURA 60 - MANTER PERFIL VOLUNTÁRIO



FONTE: Os Autores (2018).

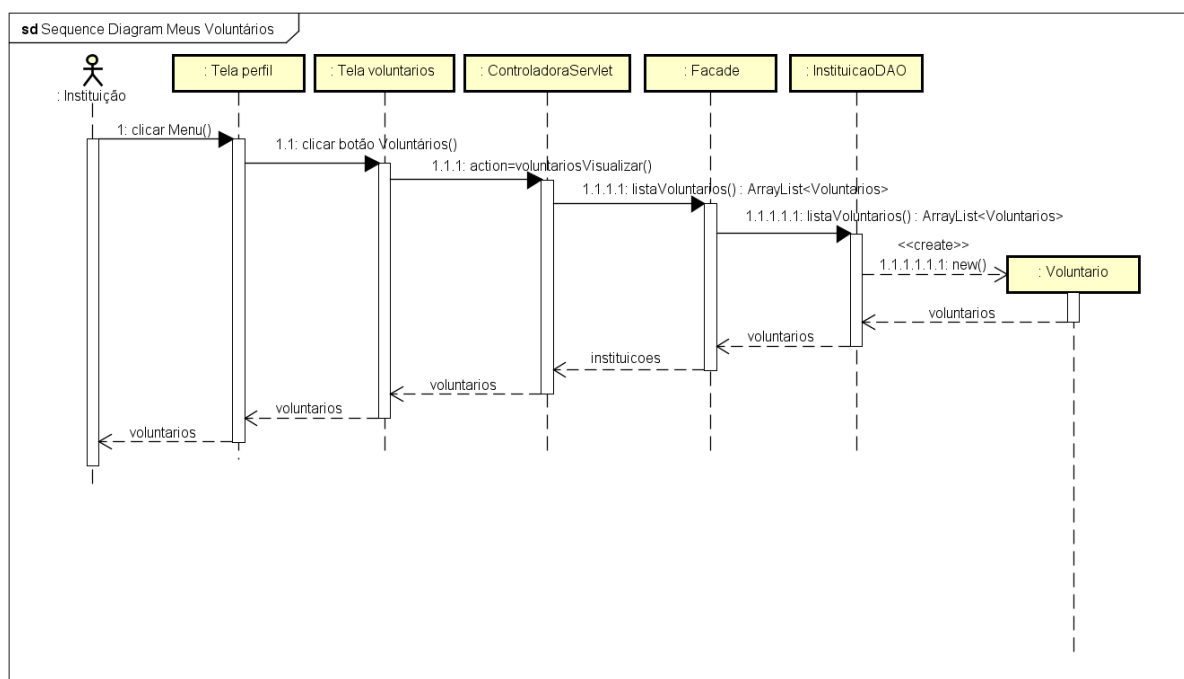


FIGURA 61 - MEUS AMIGOS



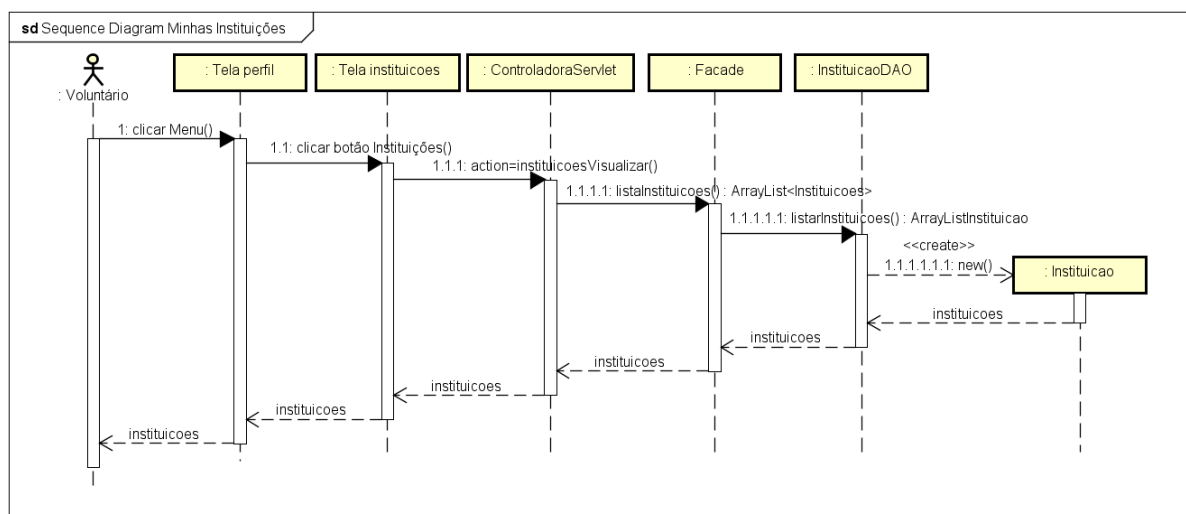
FONTE: Os Autores (2018).

FIGURA 62 - MEUS VOLUNTÁRIOS



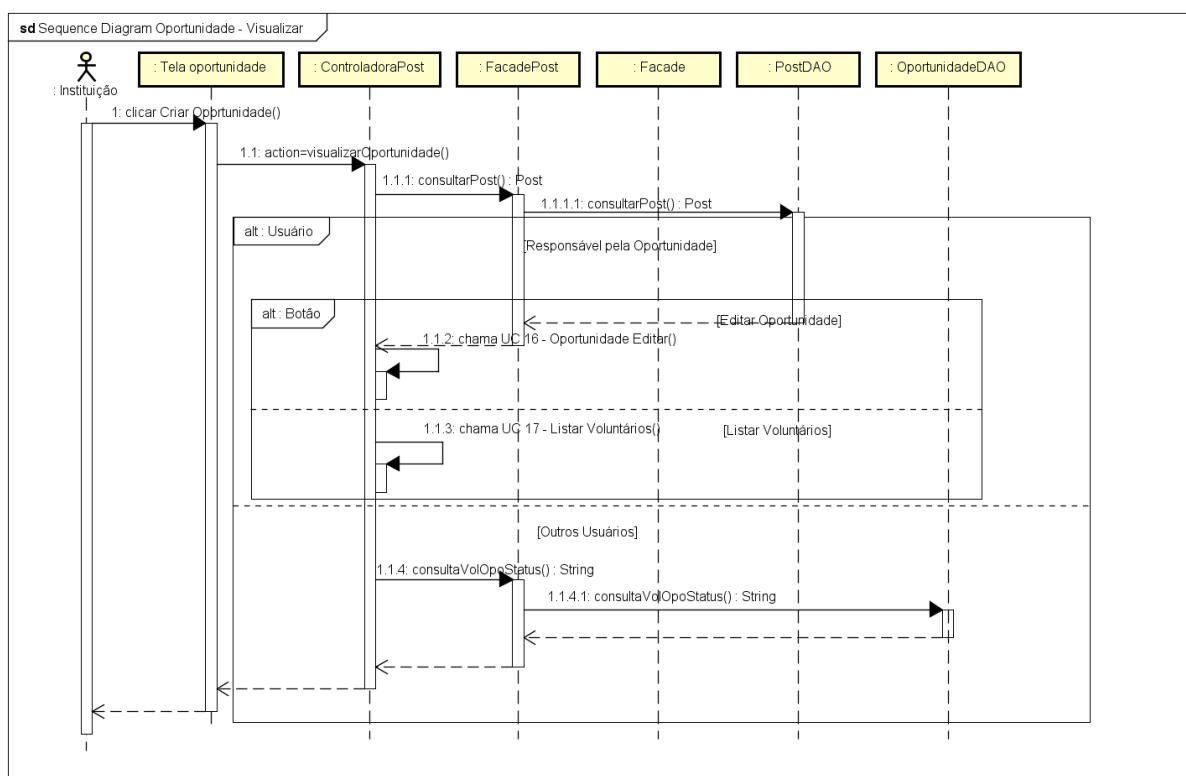
FONTE: Os Autores (2018).

FIGURA 63 - MINHAS INSTITUIÇÕES



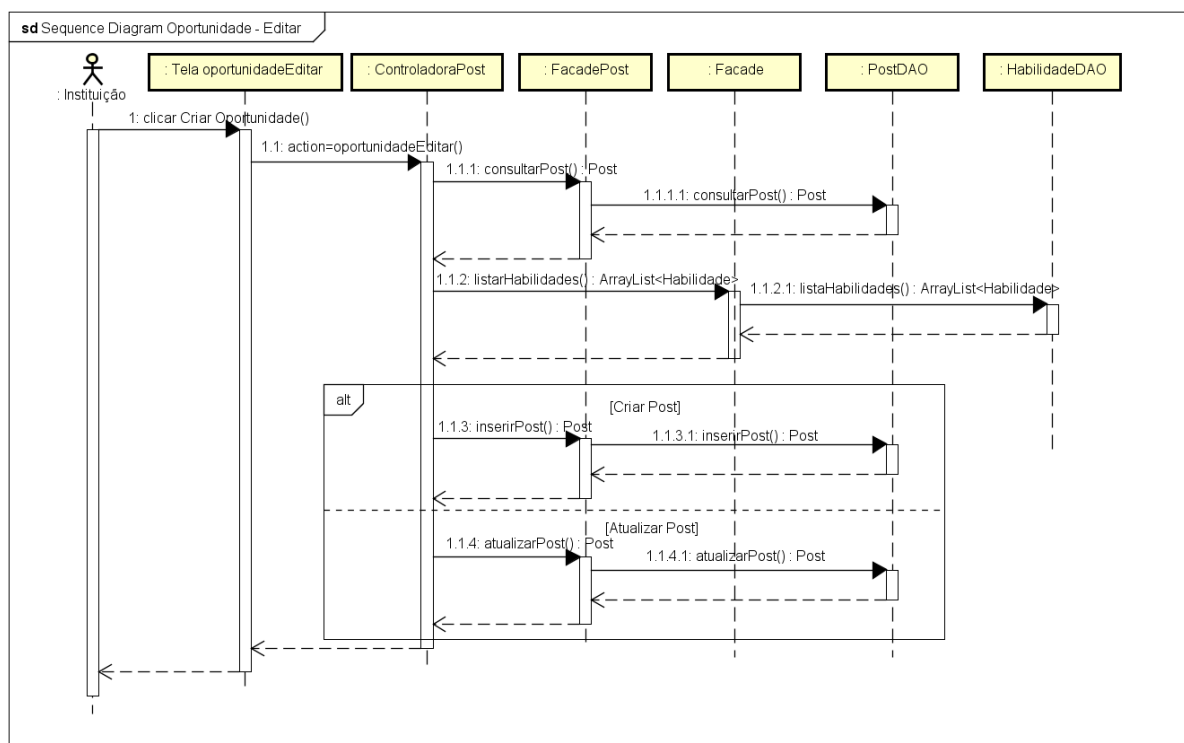
FONTE: Os Autores (2018).

FIGURA 64 - OPORTUNIDADE – VISUALIZAR



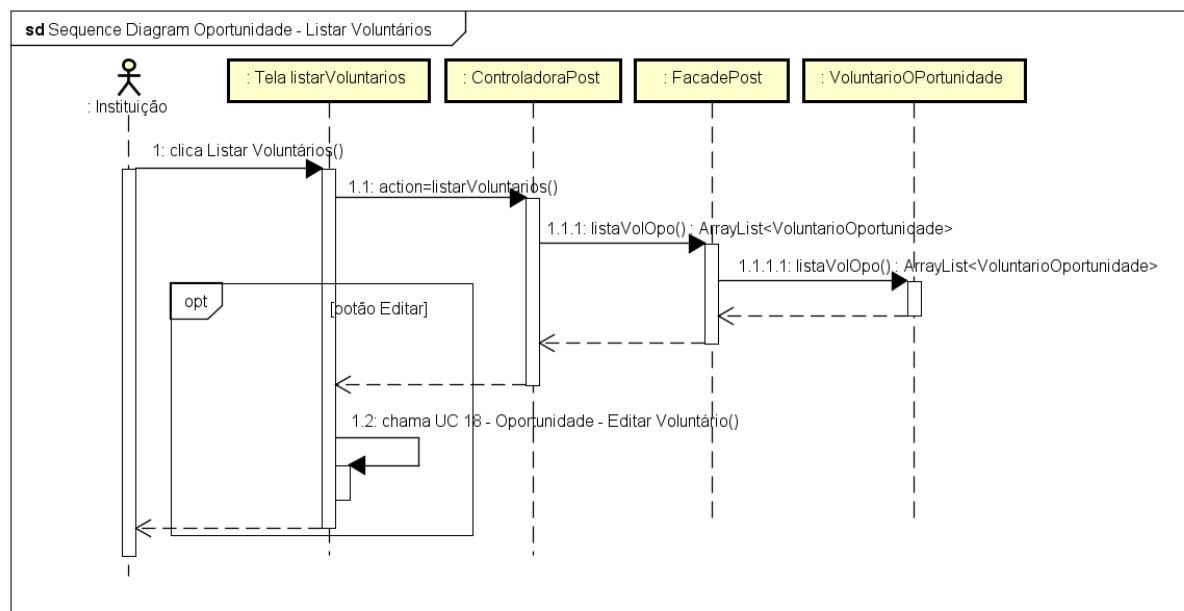
FONTE: Os Autores (2018).

FIGURA 65 - OPORTUNIDADE – EDITAR



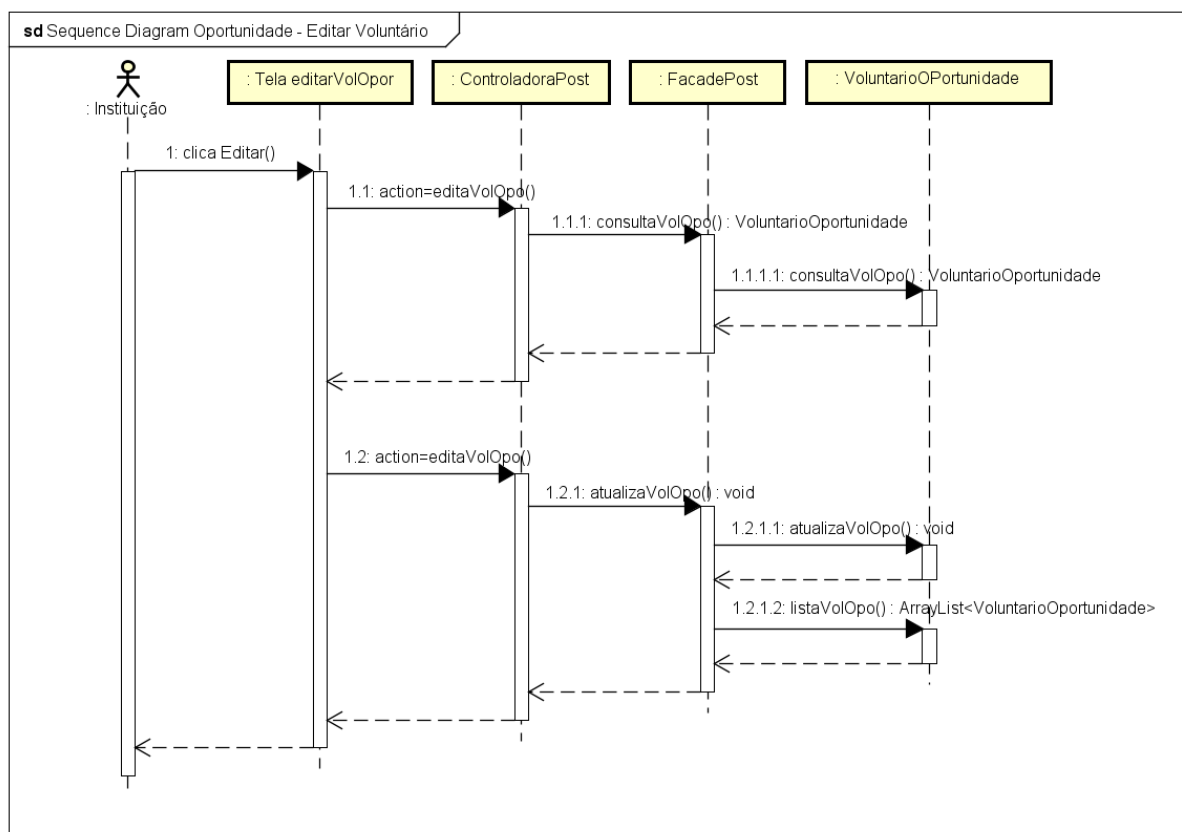
FONTE: Os Autores (2018).

FIGURA 66 - OPORTUNIDADE - LISTAR VOLUNTÁRIOS



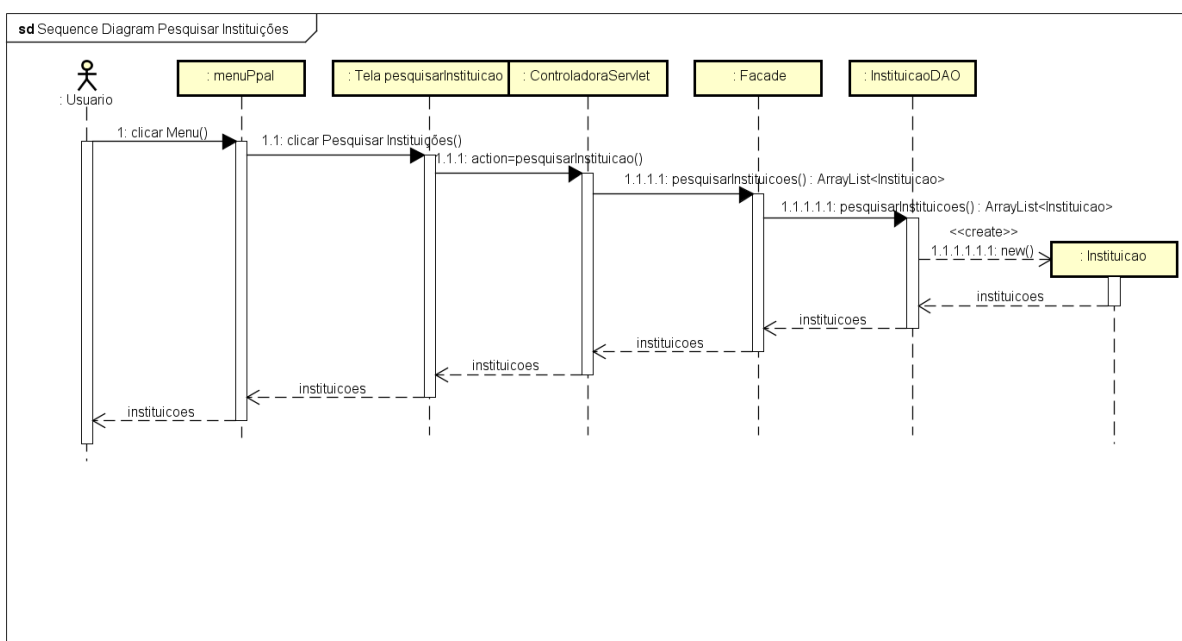
FONTE: Os Autores (2018).

FIGURA 67 - OPORTUNIDADE – EDITAR VOLUNTÁRIO



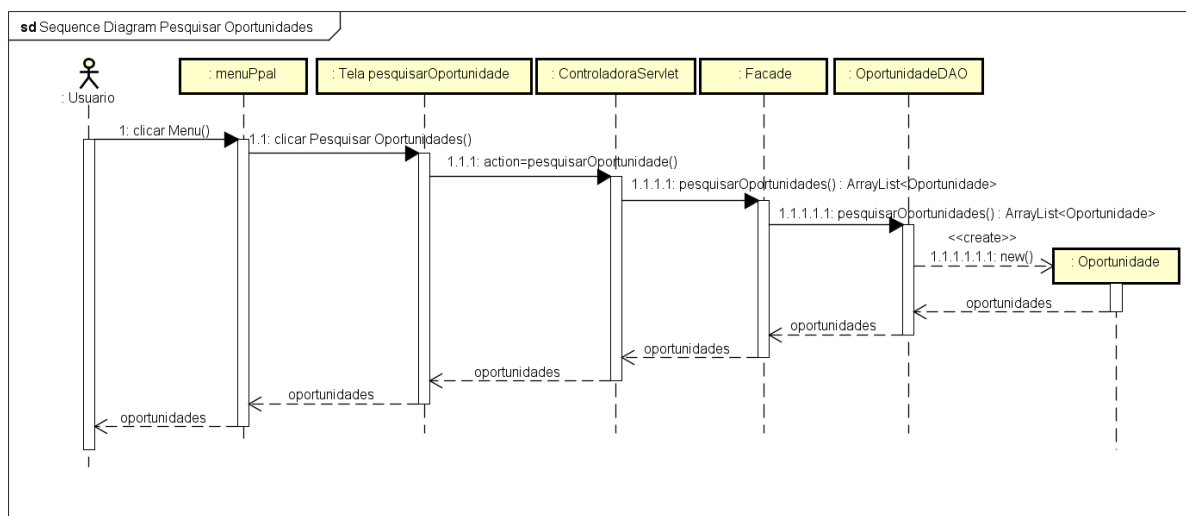
FONTE: Os Autores (2018).

FIGURA 68 - PESQUISAR INSTITUIÇÕES



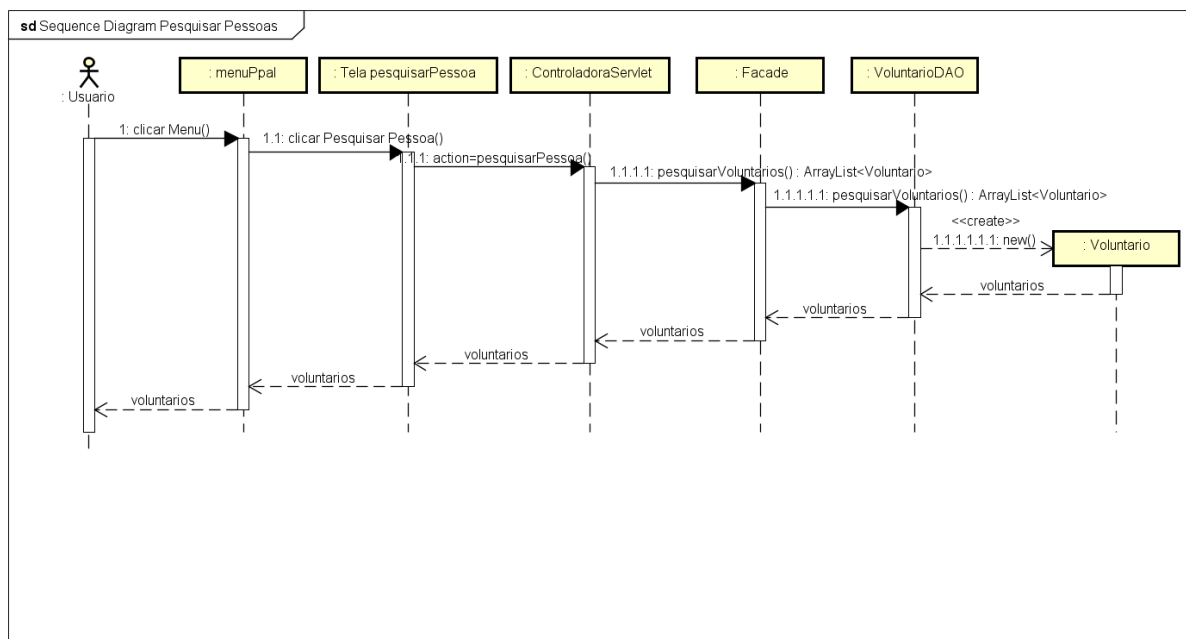
FONTE: Os Autores (2018).

FIGURA 69 - PESQUISAR OPORTUNIDADES



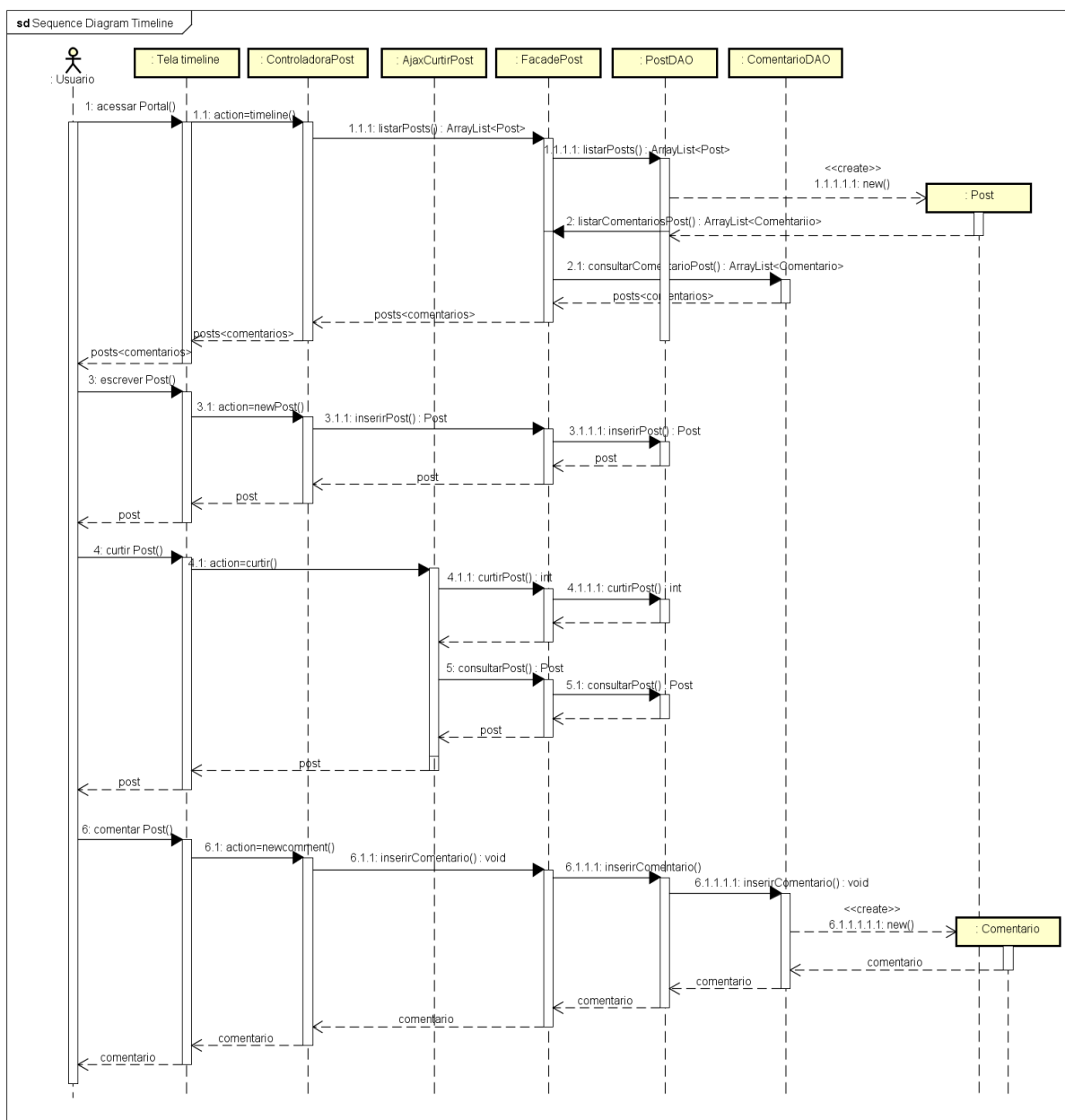
FONTE: Os Autores (2018).

FIGURA 70 - PESQUISAR PESSOAS



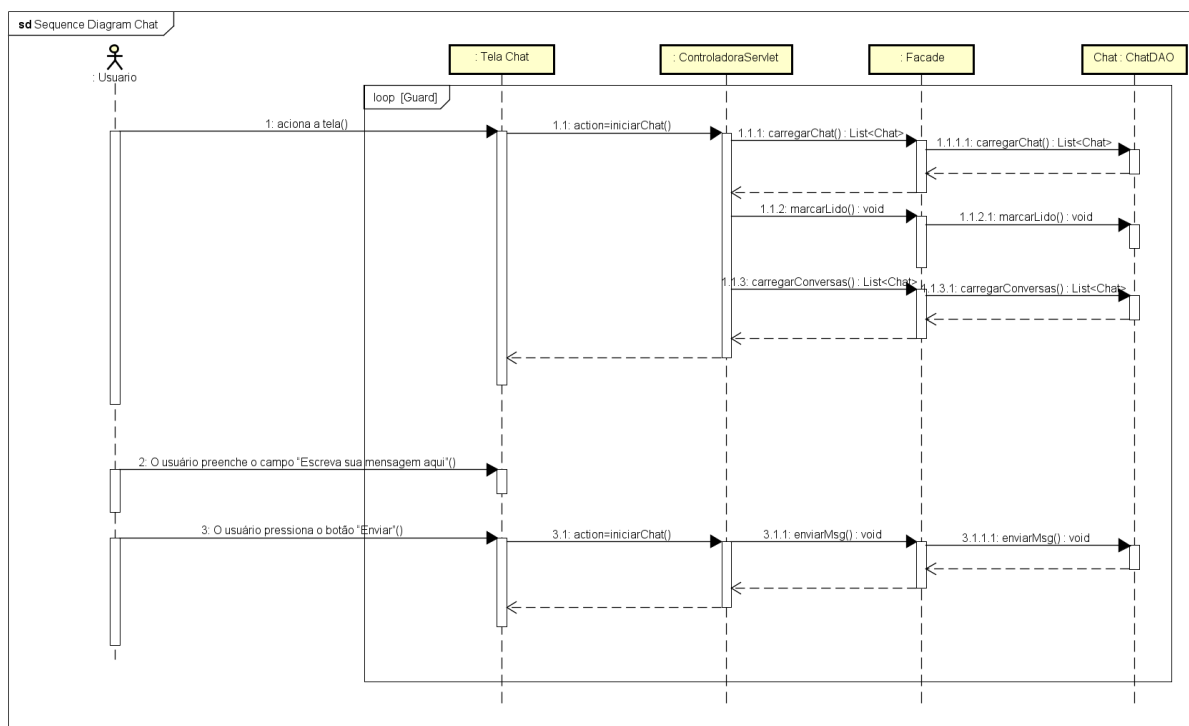
FONTE: Os Autores (2018).

FIGURA 71 - TIMELINE



FONTE: Os Autores (2018).

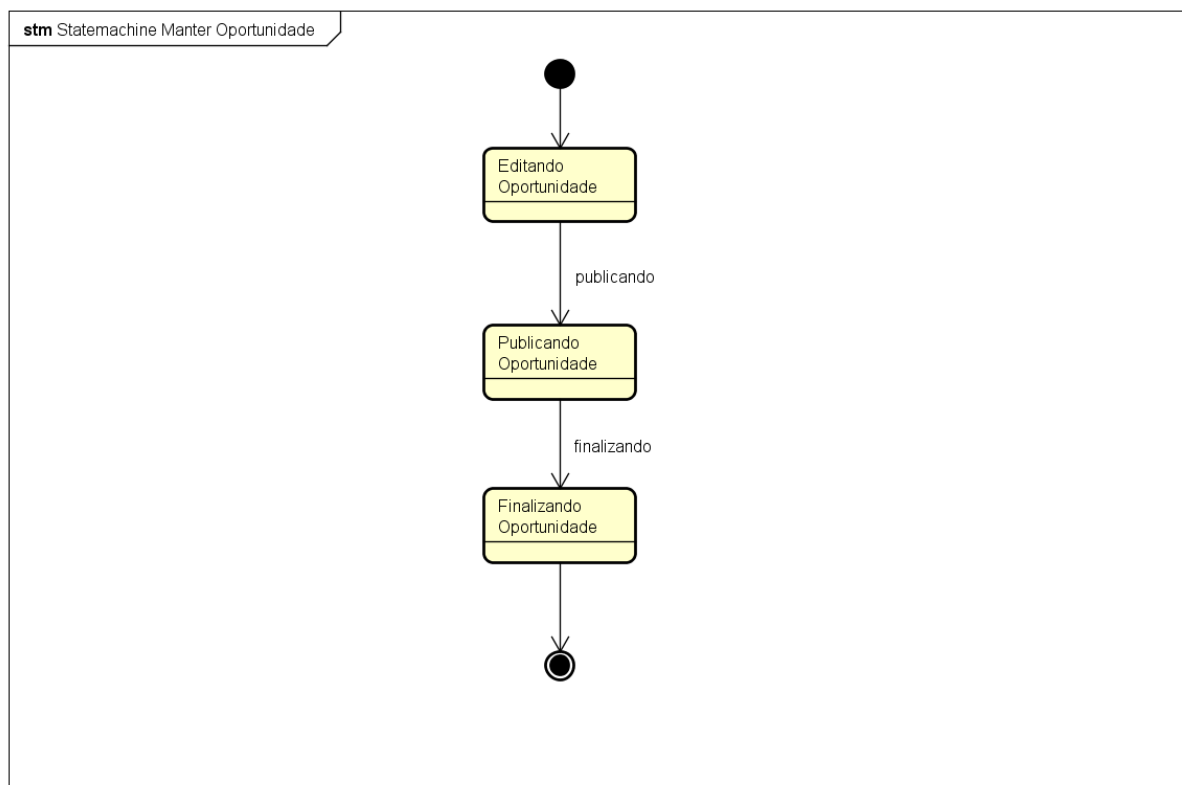
FIGURA 72 – CHAT



FONTE: Os Autores (2018).

## APENDICE G - DIAGRAMA DE ESTADO

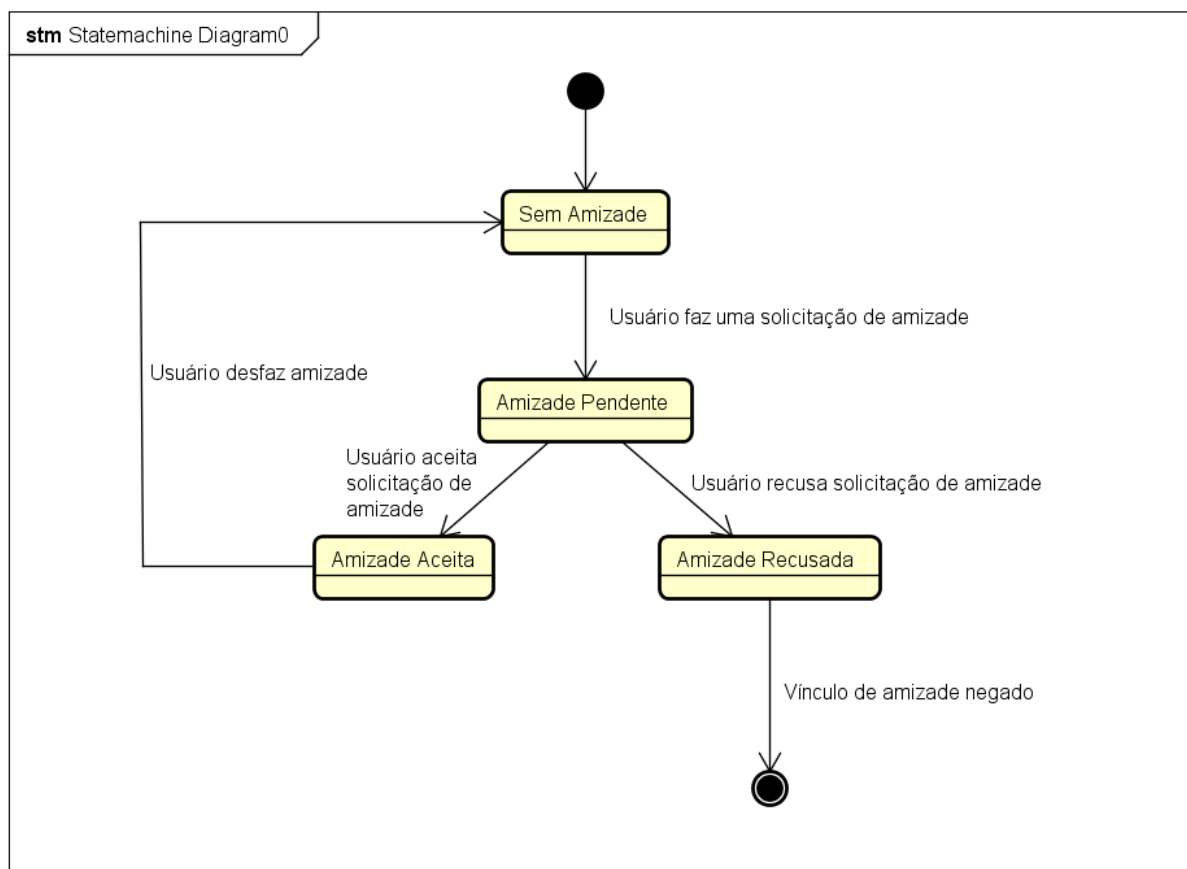
FIGURA 73 – MANTER OPORTUNIDADE



FONTE: Os Autores (2018).



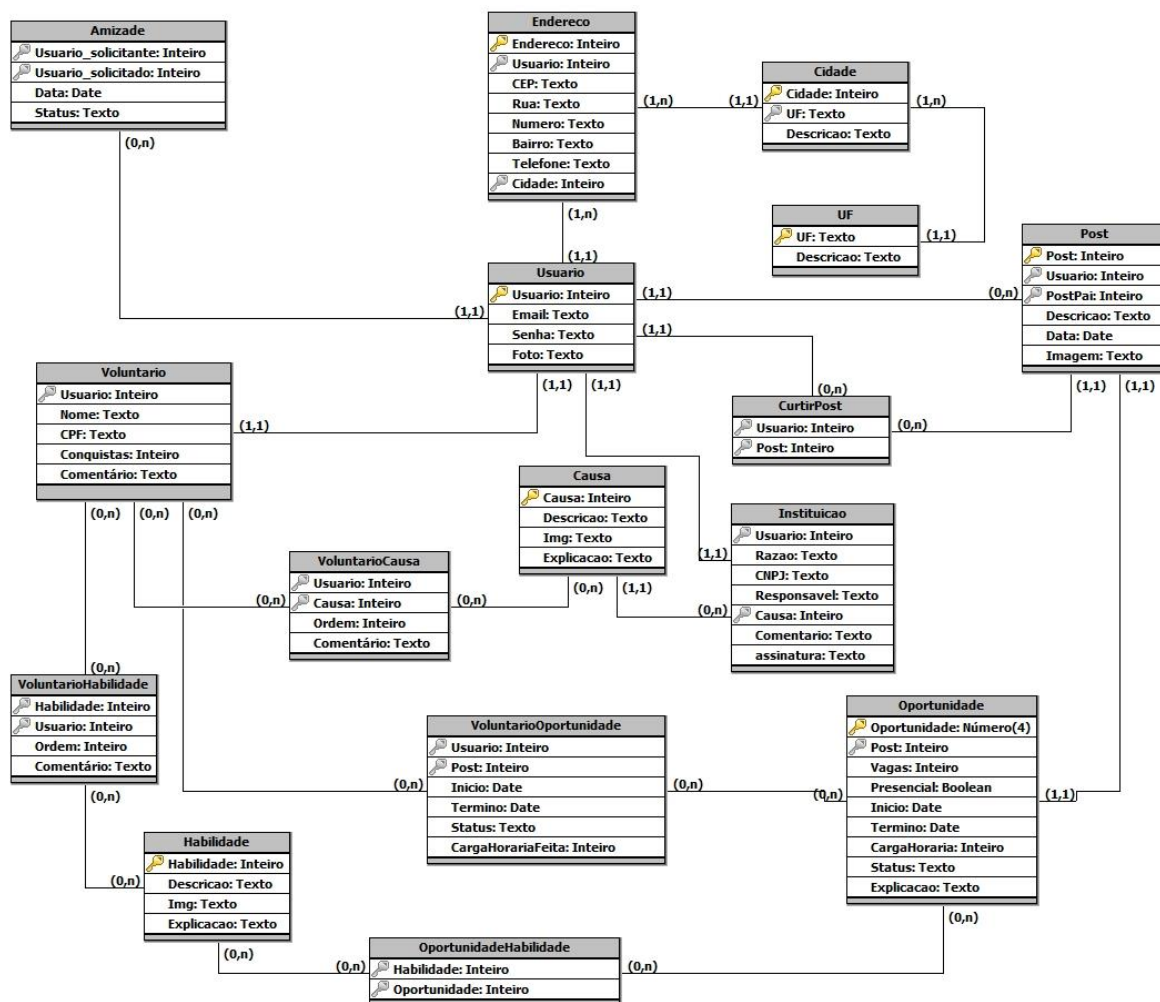
FIGURA 74 – MANTER AMIZADE



FONTE: Os Autores (2018).

## APENDICE H - DIAGRAMA DE ENTIDADE RELACIONAMENTO

FIGURA 75 - DIAGRAMA DE ENTIDADE RELACIONAMENTO



FONTE: Os Autores (2018).